

AEG



DATENVERARBEITUNG

TR 440

Große Befehlsliste

Lfd.Nr.	Datum	Seite	Befehl	Spalte	Berichtigung
---------	-------	-------	--------	--------	--------------

6	10.2.70	2		3	Bei einem Teil der letzten Auflage ist unter "12 Takte" der Text verstümmelt. Es muß in der 2. Zeile heißen: Sind unter "Werk" Befehls- und Rechenwerk aufgeführt, so gilt die Zeit für beide Werke. Tragen Sie als letzte Zeile ein: Für genaue Zeitberechnung siehe "TR 440 Befehls-Lexikon".
		4	BSS	Bemerkungen	In der 2. Zeile muß es heißen: s, ist ohne Bedeutung, muß aber angegeben werden.
		5	CNZ	Wirkung	In diese Spalte ist vor der ersten Zeile einzutragen: Voraussetzung: (A) = 0, f Bits
		6	QBR QCR }	Bemerkungen	Bitte eintragen: (nicht zugreifbares Register)
		13	PDP	Werk	Der Befehl belegt auch das Befehlswerk; bitte in dieser Spalte zusätzlich ein B eintragen.
		15	MABI	Bemerkungen	statt 2 Sternchen bitte 4 Sternchen eintragen: unter den Fußnoten einkleben: ****für c die Sprungbefehle von Seite 12 u. 13 (mit Ausnahme der Befehle NULL, PDP, PDP und WB) bei erfüllter Sprungbedingung ist ein Sprung in eine andere Großseite möglich.
		19	Zentral- code	Feld 107	In dieses Feld ist das Zeichen "inverser Schrägstrich" (\) einzutragen.
				Feld 35 u.36	Die Zeichen "T1" und "T2" sind zu tilgen; die Felder bleiben leer.
		23	TRX	im Text	Es muß in der ersten Zeile heißen: Von den Bits 9 - 12 darf höchstens eines = L sein, ...
		29 Umschlag innen	ZTO M MFU XB XC XCN RX MRX }	Code Int.	Der Befehlscode wird ZTO geschrieben (Ziffer 0 statt Buchstabe O) Auf Anregung eines Benutzers ist zu empfehlen, in der Spalte Int. (Interncode) bei diesen Befehlen ein Sternchen zu machen und als Fußnote einzukleben: *Diese Befehle benutzen gemeinsame Interncodes. Unterschiede liegen im Adressenteil siehe Seiten 22/23 "Spezifikationen-Intern".



TR 440 Große Befehlsliste

zum Programmieren von
Operatoren in der
Programmiersprache TAS

Einleitung

Dem Programmierer des Digital-Rechners TR 440 soll mit dieser Liste eine handliche Arbeitsunterlage zum Programmieren von Operatoren gegeben werden. In knapper Tabellenform stehen hier die wesentlichen Informationen zur Verfügung. Auf weitergehende Einzelheiten ist, um eine gute Übersichtlichkeit zu erzielen, verzichtet worden. Die hier nicht angegebenen Einzelheiten sind dem "TR 440-Befehlslexikon" zu entnehmen.

Diese erste Ausgabe soll erweitert werden und zu einem späteren Zeitpunkt weitere Informationen und Arbeitshilfen bieten.

Inhaltsverzeichnis	Seite
Bedeutung der Spalten	2
Erklärung der Zeichen	3
Transportbefehle	4-6
Festkomma-Arithmetik	7
Gleitkomma-Arithmetik	8
Boolesche Operationen	9
Halbwort-Arithmetik	9
Teilwort-Arithmetik	9
Index-Arithmetik	10
Setzen und Löschen	10, 11
Sprünge	12, 13
Modifizieren	14
Ersetzen (und modifizieren)	15
Aufbereitung	16, 17
Tabelle durchsuchen	17
Zentralcode	19
Potenzen von 2	20
Konvertierungstafel	21
Interncode - Externcode	22
Internspezifikationen	23
Wortstruktur	24
Blockschaltbild	25
Alphabetische Liste der Befehle	29

Bedeutung der Spalten

Bezeichnung	Code	adr	Wirkung	mod2	R	veränderliche Spalten	Alorm						
							(M)	BÜ	TK	Werk	Takte	Int.	Bemerkungen
① Erläuterungen des <u>Mnemo</u> codes	② EMC	③ n	④ Hauptwirkung	⑤ +	⑥ x	⑦	⑧ x	⑨ E≥	⑩ ≠1	⑪ B R	⑫ 22	⑬ AB	⑭ Spezielles

* ⑮
**
*** Fußnoten

- ① **Bezeichnung:** Hier steht die Befehlsbezeichnung. Die Buchstaben die zum Mnemocode führen, sind unterstrichen.
- ② **Code:** Externcode (mnemotechnische Abkürzung der Befehlsbezeichnung)
- ③ **adr:** Adressenteil des Befehls
- ④ **Wirkung:** Hier wird die hauptsächliche Wirkung des Befehls angegeben. In den veränderlichen Spalten ⑦ werden auch evtl. auftretende Nebenwirkungen angegeben.
- ⑤ **mod2:** Vom vorhergehenden Befehl kann ein Modifikator 2. Art (mod2) vorhanden sein. Es ist hier angegeben wie dieser Modifikator auf den beschriebenen Befehl einwirkt.
- + = mod2 wird während der Abrufphase zum Adressenteil des Befehls addiert. Anschließend wird mod2 gelöscht. adr := adr + mod2
mod2 := 0
- sp = Der Befehl wird während der Ausführungsphase speziell modifiziert. Die Art der Modifizierung ist bei diesen Befehlen angegeben.
- leer = Ist in der Spalte mod2 keine Eintragung, so wird der Befehl nicht modifiziert; mod2 wird in der Abrufphase gelöscht.
mod2 := 0
- ⑥ **R:** Ein x in dieser Spalte zeigt an, daß der Befehl als Zweitcode beim Registerbefehl R zugelassen ist.

- ⑦ **veränderliche Spalten:** Diese Spalten sind je nach Wirkung des Befehls unterschiedlich benannt. Im allgemeinen definieren diese Spalten den Inhalt der angegebenen Register nach Ausführung des Befehls. Bei Spalten ohne Eintragung und Registern, die nicht aufgeführt sind, erfolgt keine Änderung des betreffenden Inhaltes. Folgende Schreibweisen werden verwendet: Bei allen Registern mit Typenkennung und bei den Speicherzellen wird vor einem Semikolon die Typenkennung angegeben (t_n ; = Typenkennung der Speicherzelle n). Steht nach dem Semikolon und vor einem Komma 0 bzw. v, so wird damit ausgesagt, daß das Register links mit Null oder vorzeichengleich aufgefüllt wird (t_n ;v,m) = In dem Register, das in dieser Spalte angegeben ist, steht der Inhalt der Speicherzelle m, links mit Vorzeichen aufgefüllt; beim Transport wird die TK aus der Speicherzelle m mit in dieses Register gebracht). Ist eine Klammer $\langle \rangle$ in zwei Register-spalten eingetragen, so steht in beiden Registern der gleiche Wert.
- ⑧ **<M>:** Für das Markenregister ist durch x angegeben, daß bei diesem Befehl das Markenbit berücksichtigt wird. Als Wirkung gehört dazu, daß beim Transport eines Zahlwortes in ein Register das erste Bit dem zweiten angeglichen wird. Ist das Zahlwort markiert, so wird außerdem das Markenregister M gesetzt.
 $\langle M \rangle := \langle M \rangle V \langle n \rangle_1 \quad \langle a \rangle_1 := \langle a \rangle_2$
a = beliebiges Register
- ⑨ **BÜ-Alarm:** Hier wird angegeben, unter welchen Bedingungen ein Bereichsüberschreitungs-Alarm gegeben wird (nur bei TK 0 und 1).
> = übergelaufen \geq = über- oder untergelaufen
- ⑩ **TK-Alarm:** Diese Spalte gibt an, bei welcher Bedingung ein Typenkennungs-Alarm auftritt.

- ⑪ **Werk:** Es wird angegeben, ob der Befehl das Befehlswerk (B), das Rechenwerk (R) oder beide belegt.
- ⑫ **Takte:** Hier wird die Ausführungszeit in Takteten angegeben. Es handelt sich zum Teil um Mittelwerte. Sind unter "Werk" Befehls- und Rechenwerk aufgeführt, so gilt die Zeit für beide Werke. Für das Rechenwerk kann in vielen Fällen eine kürzere Zeit benötigt werden, die aber dann ohne Vorteil ist. Ist unter Werk nur das Rechenwerk aufgeführt, dann können während der angegebenen Zeit parallel dazu alle Befehle ablaufen, die nur das Befehlswerk ansprechen.
- Zu den angegebenen Zeiten kommen für einfache Zeitberechnungen nachfolgend diese Zeiten hinzu, die in der Abrufphase liegen und das Befehlswerk belegen:
- | Takte (Mittelwert) | | |
|--------------------|---|---|
| | Befehlsabruf | 8 |
| | belegt Rechenwerk | 1 |
| | Modifizierung 1. Art | 8 |
| | Modifizierung 2. Art (wenn bei "mod2: +") | 8 |
| | bei Operand aus dem Speicher | 8 |
| | bei erfüllter Sprungbed. | 5 |
- Für genauere Zeitberechnungen s. TR 440 Befehls-Lexikon.
- ⑬ **Int.:** Diese Spalte nennt den Interncode des Befehls in zwei Sedezimalen.
- ⑭ **Bemerkungen:** Hier sind spezielle Vermerke und Erläuterungen zum Befehl aufgeführt.
- ⑮ **Fußnoten:** Sternchen in den Spalten werden in den Fußnoten erläutert.

Erklärung der Zeichen

Bezeichnung der Register im Rechenwerk:			Zeichen und ihre Bedeutung:	Indizes für Teile eines Wortes:															
A	Akkumulator	} 48 Bits Information 2 Bits Typenkennung	:= Die links stehende Zielgröße ergibt sich aus der rechts stehenden Quellengröße Beispiel: $\langle A \rangle := \langle n \rangle$ Der Inhalt von A ergibt sich aus dem Inhalt von n.	$\langle \rangle_v$ Vorzeichen vom Inhalt															
Q	Quotientenregister			$\langle \rangle_t$ Typenkennung vom Inhalt															
D	Multiplikandenregister			$\langle \rangle_m$ Marke vom Inhalt															
H	Hilfsregister			$\langle A \rangle_1$ Bit 1 im Register A															
Y	Schifftzähler	8 Bits	::: Links- und rechtsstehende Größen werden miteinander vertauscht. Beispiel: $\langle A \rangle ::= \langle H \rangle$ Die Inhalte von A und H werden vertauscht.	$\langle A \rangle_{41-48}$ Bits 41 bis 48 im Register A															
M	Markenregister	1 Bit		$\langle n \rangle_{9-24}$ Bits 9 bis 24 in der Speicherzelle (Drittelwort)															
A,Q	doppeltlanges Register A und Q			$\langle n \rangle_{1,2}$ Bits 1 und 2 in der Speicherzelle															
H,Q	doppeltlanges Register H und Q			zur Zählung der Bits siehe Seite Wortstruktur															
Bezeichnung der Register im Befehlswerk:																			
B	Bereitadressenregister	} 24 Bits	t_x Typenkennung im Register x oder in der Speicherzelle x	Logische Verknüpfungen: \wedge UND-Verknüpfung (Konjunktion) <table><tr><td colspan="3">$a := b \wedge c$</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>L</td></tr><tr><td>0</td><td>L</td><td>0</td></tr><tr><td>L</td><td>L</td><td>L</td></tr></table>	$a := b \wedge c$			0	0	0	0	0	L	0	L	0	L	L	L
$a := b \wedge c$																			
0	0	0																	
0	0	L																	
0	L	0																	
L	L	L																	
F	Befehlsfolgeregister																		
X	Indexbasisregister	22 Bits	1,1; Typenkennung in beiden Registern eines doppeltlangen Registers																
K	Merklicherregister	} 8 Bits	0, der linke Teil des Registers ist mit Null aufgefüllt	\vee ODER-Verknüpfung (Disjunktion) <table><tr><td colspan="3">$a := b \vee c$</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>L</td><td>0</td><td>L</td></tr><tr><td>L</td><td>L</td><td>0</td></tr><tr><td>L</td><td>L</td><td>L</td></tr></table>	$a := b \vee c$			0	0	0	L	0	L	L	L	0	L	L	L
$a := b \vee c$																			
0	0	0																	
L	0	L																	
L	L	0																	
L	L	L																	
U	Unterprogrammregister																		
Variable:																			
mod1	Modifikator erster Art	} 24-Bit-Größe im Register B oder in einem nicht adressierbaren Register	$\langle \rangle$ Inhalt eines Registers, einer Speicher- oder Indexzelle Beispiel: $\langle A \rangle$ Inhalt des Registers A (einschließlich Typenkennung)	\neq Antivalenz-Verknüpfung (exklusives ODER) <table><tr><td colspan="3">$a := b \neq c$</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>L</td><td>0</td><td>L</td></tr><tr><td>L</td><td>L</td><td>0</td></tr><tr><td>0</td><td>L</td><td>L</td></tr></table>	$a := b \neq c$			0	0	0	L	0	L	L	L	0	0	L	L
$a := b \neq c$																			
0	0	0																	
L	0	L																	
L	L	0																	
0	L	L																	
mod2	Modifikator zweiter Art																		
op	Operationscode eines Befehls 8 Bits																		
adr	Inhalt des Adressenteils eines Befehls (16 Bits auf 24 Bits erweitert)																		
n	Speicheradresse eines Ganzwortes nur geradzahlige Adressen, ungeradzahlige werden um 1 vermindert.	} 16 Bits	$\langle \rangle, \langle \rangle$ Inhalt von zwei getrennten Registern Beispiel: $\langle A \rangle, \langle Q \rangle$ Inhalt von A und Q	\neg Negation <table><tr><td colspan="2">$a := \neg b$</td></tr><tr><td>0</td><td>L</td></tr><tr><td>L</td><td>0</td></tr></table>	$a := \neg b$		0	L	L	0									
$a := \neg b$																			
0	L																		
L	0																		
m	Speicheradresse eines Halbwortes																		
z	Zahl oder Operand																		
i	Indexadresse	} Indizes: L für links R für rechts Spezifikation (s evtl. unterteilt in s_1, s_2, \dots)	$ $ Betrag einer Größe, Beispiel: $ \langle n \rangle $ Betrag vom Inhalt der Speicherzelle n																
p	Parameter																		
s	Spezifikation																		
c	Zweitcode (Code für den Zweitbefehl)																		

Transportbefehle

4

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle A \rangle$	$\langle Q \rangle$	$\langle H \rangle$	$\langle D \rangle$	$\langle Y \rangle$	$\langle B \rangle$	$\langle M \rangle$	Werk	lakte	Int.	Bemerkungen
<u>Bringe</u> (nach A)	B	n	$\langle A \rangle := \langle n \rangle$	+	x	$t_n; \langle n \rangle$						x	R	2	70	
<u>Bringe</u> <u>unverändert</u>	BU	n	$\langle A \rangle := \langle n \rangle$	+	x	$t_n; \langle n \rangle$							R	2	D3	unverändert bei jeder TK
<u>Bringe</u> und <u>reserviere</u>	BR	n	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := \langle n \rangle$	+	x	$t_n; \langle n \rangle$		$t_A; \langle A \rangle$				x	R	5	76	
<u>Bringe</u> <u>negativ</u>	BN	n	$\langle A \rangle := -\langle n \rangle$	+	x	$t_n; -\langle n \rangle$						x	R	5	75	
<u>Bringe</u> <u>negativ</u> und <u>reserviere</u>	BNR	n	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := -\langle n \rangle$	+	x	$t_n; -\langle n \rangle$		$t_A; \langle A \rangle$				x	R	5	77	
<u>Bringe</u> <u>Betrag</u>	BB	n	$\langle A \rangle := \langle n \rangle $	+	x	$t_n; \langle n \rangle $						x	R	5	74	
<u>Bringe</u> nach <u>Q</u>	BQ	n	$\langle Q \rangle := \langle n \rangle$	+	x		$t_n; \langle n \rangle$					x	R	3	72	
<u>Bringe</u> nach <u>D</u>	BD	n	$\langle D \rangle := \langle n \rangle$	+	x				$t_n; \langle n \rangle$			x	R	3	71	
<u>Bringe</u> nach <u>H</u>	BH	n	$\langle H \rangle := \langle n \rangle$	+	x			$t_n; \langle n \rangle$				x	R	3	73	
<u>Bringe</u> nach <u>Q</u> und <u>bringe</u> (nach A)	BQB	n	$\langle A \rangle := \langle n \rangle$ $\langle Q \rangle := \langle n \rangle$	+	x	$t_n; \langle n \rangle$	$t_n; \langle n \rangle$					x	R	2	DA	
<u>Bringe</u> <u>zwei</u> <u>Wörter</u>	BZ	n	$\langle A \rangle := \langle n \rangle$ $\langle Q \rangle := \langle n+2 \rangle$	+		$t_n; \langle n \rangle$	$t_{n+2}; \langle n+2 \rangle$					x	B R	15	D9	
<u>Bringe</u> <u>zwei</u> <u>Wörter</u> <u>negativ</u>	BZN	n	$\langle A \rangle := -\langle n \rangle$ $\langle Q \rangle := -\langle n+2 \rangle$	+		$t_n; -\langle n \rangle$	$t_{n+2}; -\langle n+2 \rangle$					x	B R	15	D1	
<u>Bringe</u> <u>Teilwort</u>	BT	n	$\langle A \rangle_x := \langle n \rangle_x$ falls $\langle Q \rangle_x = 0$ $\langle A \rangle_x := 0$ falls $\langle Q \rangle_x = L$	+	x	$t_n; \dots$ (s. Wirk.) $\langle A \rangle$ wird um p Stellen rechts geschiftet, 0 nachgezogen							R	7 +2p	F6	p: Anzahl der L, die rechtsbündig in Q stehen. x: 1,2,...,48
<u>Bringe</u> <u>nächstes</u> <u>Zeichen</u>	BNZ	it i_R	$\langle A \rangle := 0$, Zeichen gemäß a und b rechtsb. $\langle A \rangle_t := \langle a \rangle_t$ $b := b + 1$ wenn $b < d$ $b := +0$ $\langle i_R \rangle := \langle i_R \rangle + 2$ wenn $b = d$		sp	$t_n; 0, \dots$ (s. Wirk.)			undefiniert		$\langle i_L \rangle_{alt}$		B R	41	E6	a: $\langle i_R \rangle + \text{mod2}$ lauf. Adr. eines Wortes der Liste b: $\langle i_L \rangle_{17-24}$ lauf. Nummer eines Zeichens im Wort (0,1,...) d: $(48/f)-1$ max. Zeichennummer f: $\langle i_L \rangle_{9-12}$ Anzahl der Bits pro Zeich. (4,6,8 oder 12) andere Bits von $\langle i_L \rangle$ bedeutungslos
<u>Bringe</u> <u>zwei</u> <u>Halbwörter</u>	BZ2	m	$\langle A \rangle := \langle m, m+1 \rangle$	+		$t_n; \langle m, m+1 \rangle$							B R	10	D8	
<u>Bringe</u> <u>Halbwort</u>	B2	m	$\langle A \rangle := \langle m \rangle$	+	x	$t_n; 0, \langle m \rangle$							R	9	6E	
<u>Bringe</u> <u>Halbwort</u> mit <u>Vorzeichen</u>	B2V	m	$\langle A \rangle := \langle m \rangle$	+	x	1; v, $\langle m \rangle$							R	10	6F	
<u>Bringe</u> <u>Halbwort</u> mit <u>Vorzeichen</u> <u>negativ</u>	B2VN	m	$\langle A \rangle := -\langle m \rangle$	+	x	1; -v, $\langle m \rangle$							R	12	67	
<u>Bringe</u> <u>Drittelwort</u>	B3	m	$\langle A \rangle := \langle m \rangle_{9-24}$	+	x	1; 0, $\langle m \rangle_{9-24}$							R	9	6C	
<u>Bringe</u> <u>Drittelwort</u> mit <u>Vorzeichen</u>	B3V	m	$\langle A \rangle := \langle m \rangle_{9-24}$	+	x	1; v, $\langle m \rangle_{9-24}$							R	10	6D	

<u>Bringe</u> aus <u>Leitblock</u>	BLEI	p	$\langle A \rangle := \langle \langle BL \rangle \cdot 2^p + p \rangle$	+		$t; \langle \langle BL \rangle \cdot 2^p + p \rangle$							B R	13	BK	p: +0...255 BL = Leitadressenregister
<u>Bringe</u> (und <u>setze</u>) <u>Steuerbits</u> (und <u>Sperren</u>)	BSS	s_L s_R	$\langle A \rangle := \text{Steuerbits}$ falls $a_0 = L$			2; Steuerbits falls $a_0 = L$							B R	3	FB	a_L : Bits a_0 bis a_{16} s_R : Bits a_{17} bis a_{24}

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle n \rangle$	$\langle n \rangle$	$\langle n+2 \rangle$	$\langle A \rangle$	$\langle M \rangle$	BÜ	TK	Werk	Iakte	Int.	Bemerkungen
Bringe und lösche	BL	n	$\langle A \rangle := \langle n \rangle$ $\langle n \rangle := +0$	+		0;0			$t_n; \langle n \rangle$				B R	17	B0	unverändert bei jeder TK
Bringe und speichere	BC	n	$\langle A \rangle := \langle n \rangle$	+		$\langle n \rangle$	$t_n; \langle A \rangle$		$t_n; \langle n \rangle$	x	≥		B R	13	A3	

										$\langle X \rangle$	$\langle U \rangle$					
Bringe und speichere Indexbasis	BCI	n	$\langle X \rangle := \langle n \rangle_{3-24}$ $\langle U \rangle := \langle n \rangle_{41-48}$				$3; 0, \langle X \rangle, 0, \langle U \rangle$			$\langle n \rangle_{3-24}$	$\langle n \rangle_{41-48}$		B	17x +66	B6	x = Anzahl der zurückge- speicherten Indexreg.
Speichere	C	n	$\langle n \rangle := \langle A \rangle$	+	0	$t_n; \langle A \rangle$						≥	B R	8	80	
Speichere unverändert	CU	n	$\langle n \rangle := \langle A \rangle$	+	$\langle A \rangle_1$	$t_n; \langle A \rangle$							B R	8	D0	unverändert bei jeder TK
Speichere und bringe Reserve	CR	n	$\langle n \rangle := \langle A \rangle$ $\langle A \rangle := \langle H \rangle$	+	0	$t_n; \langle A \rangle$		$t_H; \langle H \rangle$				≥	B R	9	81	
Speichere negativ	CN	n	$\langle n \rangle := -\langle A \rangle$	+	0	$t_n; -\langle A \rangle$						≥	B R	9	84	
Speichere Betrag	CB	n	$\langle n \rangle := \langle A \rangle $	+	0	$t_n; \langle A \rangle $						≥	B R	9	85	
Speichere markiert	CMT	n	$\langle n \rangle := \langle A \rangle$	+	L	$t_n; \langle A \rangle$						≥	2,3 B R	8	82	
Speichere mit Marke aus Register	CMR	n	$\langle n \rangle := \langle A \rangle$	+	$\langle M \rangle$	$t_n; \langle A \rangle$						≥	2,3 B R	8	83	
Speichere mit Marke aus Speicher	CMC	n	$\langle n \rangle := \langle A \rangle$	+	$\langle n \rangle$	$t_n; \langle A \rangle$						≥	2,3 B R	13	A2	
Speichere aus Q	CQ	n	$\langle n \rangle := \langle Q \rangle$	+	0	$t_Q; \langle Q \rangle$						≥	B R	9	87	
Speichere aus D	CD	n	$\langle n \rangle := \langle D \rangle$	+	0	$t_D; \langle D \rangle$						≥	B R	9	86	
Speichere aus H	CH	n	$\langle n \rangle := \langle H \rangle$	+	0	$t_H; \langle H \rangle$						≥	B R	9	8F	
Speichere zwei Wörter	CZ	n	$\langle n \rangle := \langle A \rangle$ $\langle n+2 \rangle := \langle Q \rangle$	+	0	$t_n; \langle A \rangle$		$t_Q; \langle Q \rangle$				≥	B R	16	DB	

						Wirkung	$\langle D \rangle$									
Speichere Teilwort	CT	n	$\langle n \rangle_x := \langle A \rangle_x$	+		$\langle A \rangle$ um p Stellen links im Kreis geschiftet $\langle n \rangle_x := \langle A \rangle_x$ für $\langle Q \rangle_x = 0$ $\langle n \rangle_x := \langle n \rangle_x$ für $\langle Q \rangle_x = 1$ $\langle A \rangle$ im Kreis zurückgeschiftet	$t_n; \langle n \rangle$			B R	22 +2p	F7				p: Anzahl der L, die rechts- bündig im Reg. Q stehen x: 1,2,...,48

						$\langle A \rangle$	$\langle n \rangle$	$\langle B \rangle$	$\langle Y \rangle$	$\langle Q \rangle$	$\langle D \rangle$					
Speichere nächstes Zeichen	CNZ	i_L i_R	rechte f Bits von $\langle A \rangle$ in a gemäß b eingesetzt $\langle A \rangle := \langle a \rangle$ $\langle A \rangle_t := \langle a \rangle_t$ $b := b + 1$ wenn $b < d$ $b := +0$ $\langle i_R \rangle := \langle i_R \rangle + 2$ } wenn $b = d$	sp		$t_n; \langle a \rangle$		$\langle i_L \rangle_{a1}$	undef.	abzusp. Zeichen aus A, gem.b geschif- tet; andere Bits=0	undef.		B R	55	E7	a: $\langle i_R \rangle + \text{mod2}$ lauf. Adr. eines Wortes der Liste b: $\langle i_L \rangle_{17-24}$ lauf. Nummer eines Zeichens im Wort (0,1,...) d: $(48/f) - 1$ max. Zeichennummer f: $\langle i_L \rangle_{9-12}$ Anzahl der Bits pro Zeich. (4,6,8 oder 12) andere Bits $\langle i_L \rangle$ bedeutungslos

						$\langle m \rangle$										
Speichere Halbwort	C2	m	$\langle m \rangle := \langle A \rangle_{25-48}$	+		$\langle A \rangle_{25-48}$							B R	15	A0	Das Halbwort wird unverän- dert eingesetzt

						$\langle m \rangle_{8-24}$										
Speichere Drittelwort	C3	m	$\langle m \rangle_{8-24} := \langle A \rangle_{33-48}$	+		$\langle A \rangle_{33-48}$							B R	15	A1	Das Drittelwort wird unver- ändert eingesetzt

Transportbefehle (Fortsetzung)

Bezeichnung	Code	adr	Wirkung	mod2	R	(B)	(i)	(i _L)	(i _R)	(s ₁)	(m)	(K)	Werk	Take	Int.	Bemerkungen
Registertausch	RT	s	$\langle s_1 \rangle := \langle s_2 \rangle$	+			$\langle s_1 \rangle$ und $\langle s_2 \rangle$ werden vertauscht einschl. TK						R	4	97	$s_1, s_2: A, Q, D$ oder H; $s_1 \neq s_2$
Index: Bringe	XB	i	$\langle B \rangle := \langle i \rangle$				$\langle i \rangle$						B	1	08	Bits 9 - 11 = OLO
Index: Speichere	XC	i	$\langle i \rangle := \langle B \rangle$					$\langle B \rangle$					B	3	18	Bit 9 = 0
Index: Speichere negativ	XCN	i	$\langle i \rangle := -\langle B \rangle$					$-\langle B \rangle$					B	5		Bit 9 = L
Tausch-Transport in Indexzellen	TTX	i _L i _R	$\langle B \rangle := \langle i_L \rangle$ $\langle i_R \rangle := \langle i_L \rangle$				$\langle i_L \rangle$		$\langle i_R \rangle$	$\langle i_L \rangle$			B	11	0D	
Transport aus Indexzelle nach Indexzelle	TXX	i _L i _R	$\langle i_L \rangle := \langle i_R \rangle$ $\langle B \rangle := \langle i_R \rangle$				$\langle i_R \rangle$		$\langle i_R \rangle$				B	4	0C	
Transport aus Indexzelle nach Rechenwerk	TXR	s i	$\langle s_1 \rangle := \pm \langle i \rangle$ $\langle B \rangle := \pm \langle i \rangle$				$\pm \langle i \rangle$				$1; v, \pm \langle i \rangle$		B R	5	8C	$s_1: A, Q, D$ und H $s_2: \text{leer: } + \}$ statt \pm N : -
Transport aus Rechenwerk nach Indexzelle	TRX	s i	$\langle i \rangle := \pm \langle s_1 \rangle_{25-48}$ $\langle B \rangle := \pm \langle s_1 \rangle_{25-48}$				$\pm \langle s_1 \rangle_{25-48}$						B R	6	9C	$s_1: A, Q, D, H$ oder leer $s_2: \text{leer: } + \}$ statt \pm N : -
Transport aus Speicher nach B	TCB	m	$\langle B \rangle := \langle m \rangle$	+	x		$\langle m \rangle$						B	2	39	
Transport aus B nach Speicher	TBC	m	$\langle m \rangle := \langle B \rangle$								$\langle B \rangle$		B	7	07	
Bringe und speichere Merklichter	BCL	m	$\langle m \rangle_{17-24} := \langle K \rangle$								$\langle K \rangle$	$\langle m \rangle_{17-24}$	B	18	06	$\langle m \rangle_{1-16}$ bleiben erhalten

Voraussetzung

Wortgruppentransport vorwärts	WTV	i _L i _R	$\langle \langle i_L \rangle + 2k \rangle := \langle \langle i_R \rangle + 2k \rangle$ $\langle B \rangle := \langle i_L \rangle + 2(\langle B \rangle - 1)$				$\langle B \rangle$ = Länge der Wortgruppen in Ganzwörtern = a $\langle i_R \rangle$ = niederste Adresse des Quellenbereichs $\langle i_L \rangle$ = niederste Adresse des Zielbereichs						B	21a +10	22	k; 0, 1, 2, 3, ..., $\langle B \rangle - 1$ a: Anzahl der Ganzwörter
Wortgruppentransport rückwärts	WTR	i _L i _R	$\langle \langle i_L \rangle - 2k \rangle := \langle \langle i_R \rangle - 2k \rangle$ $\langle B \rangle := \langle i_L \rangle - 2(\langle B \rangle - 1)$				$\langle B \rangle$ = Länge der Wortgruppen in Ganzwörtern = a $\langle i_R \rangle$ = höchste Adresse des Quellenbereichs $\langle i_L \rangle$ = höchste Adresse des Zielbereichs						B	21a +10	23	

Wirkung

Bequemes Bringen aller Register	QBR	n	Die Register werden unverändert gebracht				$\langle T \rangle := \langle n \rangle_{1-24}$ $\langle M \rangle := L$, falls $\langle n-10 \rangle_t = 1$ oder 3 $\langle H \rangle := t_{n-2}; \langle n-2 \rangle$ $\langle M \rangle := 0$, falls $\langle n-10 \rangle_t = 0$ oder 2 $\langle D \rangle := t_{n-4}; \langle n-4 \rangle$ $\langle B \rangle := \langle n-10 \rangle_{1-24}$ $\langle Q \rangle := t_{n-6}; \langle n-6 \rangle$ $\langle K \rangle := \langle n-10 \rangle_{25-32}$ $\langle A \rangle := t_{n-8}; \langle n-8 \rangle$ $\langle Y \rangle := \langle n-10 \rangle_{33-40}$ $\langle U \rangle := \langle n-10 \rangle_{41-48}$						B R	84	FE	T: Prüfregister
Bequemes Speichern aller Register	QCR	n	Die Register werden unverändert abgespeichert				$\langle n \rangle_t := 2 + \langle M \rangle$ $\langle n+2 \rangle := t_A; \langle A \rangle$ $\langle n \rangle_{1-24} := \langle B \rangle$ $\langle n+4 \rangle := t_Q; \langle Q \rangle$ $\langle n \rangle_{25-32} := \langle K \rangle$ $\langle n+6 \rangle := t_D; \langle D \rangle$ $\langle n \rangle_{33-40} := \langle Y \rangle$ $\langle n+8 \rangle := t_H; \langle H \rangle$ $\langle n \rangle_{41-48} := \langle U \rangle$ $\langle n+10 \rangle := 2; \langle T \rangle, 0$						B R	55	FF	T: Prüfregister

Spezifikationen

Zeichenkettenverarbeitung			$\langle A \rangle$ bzw. $\langle A, Q \rangle := p$ Oktaden gem. a; ggf. verkürzen zu Tetraden				$s_1: \text{leer}/B$ = Speichern/Bringen $s_2: \text{leer}/I$ = Oktadenadr. nicht erhöhen/erhöhen um p (Inkrement) $s_3: \text{leer}/V$ = nicht verändern/verändern (Oktaden—Tetraden) $s_4: \text{leer}/R$ = Oktadenadresse in Indexzelle/oder Register $s_5: p$ = Anzahl der Oktaden (1...12) i: Indexadresse wenn $s_4 = \text{leer}$; Reg. D, H od. B wenn $s_4 = R$						B R	min. 122	FC	$a = \langle i \rangle$ wenn $s_4 = \text{leer}$ $a = \langle D \rangle_{25-48}$ wenn $i = D$ und $a = \langle H \rangle_{25-48}$ wenn $i = H$ $a = \langle B \rangle$ wenn $i = B$ $s_4 = R$ a: Oktadenadresse
Version Erbringen	ZK	s i	$\langle a \rangle := \langle A \rangle$ bzw. $\langle A, Q \rangle p$ Okt. ggf. verlängern aus Tetraden													
Version Speichern																

Transportiere Oktaden wenn $\langle H \rangle_1 = 3$ wenn $\langle H \rangle_1 = 2$	TOK	z	$\langle a+x \rangle := \langle q+x \rangle$ Quellgeb. z Okt. (q+6) > a $\langle a+x \rangle := \langle q+y \rangle$ Quellgeb. 6 Okt. i. ein Ganzwort q = 3n+0	3; Inhalt d. letzten Ganzwortes im Zielgebiet 3; Inhalt d. letzten Ganzwortes im Zielgebiet 2; (a+z), q	3; 0	letzte Halbw. adr. i. Zielg.	x	B R	min. 191 min. 209	FD	z: 1...65535 a: $\langle H \rangle_1 - 24 = \text{Okt. adr. Zielgebiet}$ q: $\langle H \rangle_{28-40} = \text{Okt. adr. Quellgebiet}$ x: 0, 1, 2, ..., z-1 y: 0, 1, 2, 3, 4, 5, 0, 1, 2, ...
---	-----	---	--	---	------	------------------------------	---	-----	----------------------	----	---

Festkomma-Arithmetik

Bezeichnung	Code	adr	Wirkung	mod2	R	A	Q	D	n	Y	AM	BÜ	TK	Werk	Takte	Int.	Bemerkungen
Addiere	A	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle$	+	x	$t_{xxx}; \langle A \rangle + \langle n \rangle$		$t_n; \langle n \rangle$		x	≥			R	8	42	
Addiere Betrag	AB	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle $	+	x	$t_{xxx}; \langle A \rangle + \langle n \rangle $		$t_n; \langle n \rangle$		x	≥			R	8	40	
Addiere im Speicher	AC	n	$\langle n \rangle := \langle n \rangle + \langle A \rangle$	+				$t_{xxx}; \langle n \rangle + \langle A \rangle$	$t_{xxx}; \langle n \rangle + \langle A \rangle$	x	≥			B R	17	43	
Subtrahiere	SB	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle$	+	x	$t_{xxx}; \langle A \rangle - \langle n \rangle$		$t_n; \langle n \rangle$		x	≥			R	8	46	
Subtrahiere Betrag	SBB	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle $	+	x	$t_{xxx}; \langle A \rangle - \langle n \rangle $		$t_n; \langle n \rangle$		x	≥			R	8	41	
Subtrahiere im Speicher	SBC	n	$\langle n \rangle := \langle n \rangle - \langle A \rangle$	+				$t_{xxx}; \langle n \rangle - \langle A \rangle$	$t_{xxx}; \langle n \rangle - \langle A \rangle$	x	≥			B R	17	47	
Subtrahiere invers	SBI	n	$\langle A \rangle := \langle n \rangle - \langle A \rangle$	+	x	$t_{xxx}; \langle n \rangle - \langle A \rangle$		$t_n; \langle n \rangle$		x	≥			R	8	44	
Subtrahiere von D	SD	n	$\langle A \rangle := \langle D \rangle - \langle n \rangle$	+	x	$t_{xxx}; \langle D \rangle - \langle n \rangle$				x	≥			R	11	45	
Multipliziere mit Rundung	MLR	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$	+	x	$1; (\langle A \rangle \cdot \langle n \rangle)^*$	1; +0	$t_n; \langle n \rangle$		+0	x	≠1		R	57	55	
Multipliziere negativ mit Rundung	MNR	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle$	+	x	$1; (-\langle A \rangle \cdot \langle n \rangle)^*$	1; +0	$t_n; \langle n \rangle$		+0	x	≠1		R	57	59	
Multipliziere akkumulierend mit Rundung	MAR	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	x	$1; (\langle A \rangle \cdot \langle n \rangle)^* + \langle H \rangle$	1; +0	$t_n; \langle n \rangle$		+0	x	≥	≠1	R	67	57	
Multipliziere akkumulierend negativ m. Rund.	MANR	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	x	$1; (-\langle A \rangle \cdot \langle n \rangle)^* + \langle H \rangle$	1; +0	$t_n; \langle n \rangle$		+0	x	≥	≠1	R	67	58	
Dividiere	DV	n	$\langle A \rangle := \langle A \rangle : \langle n \rangle$	+	x	$1; \langle A \rangle : \langle n \rangle$	1; Rest · 2 ⁴⁸	1; +0		+0	x	**	≠1	R	220	60	Operanden, Quotient und Rest sind als echte Brüche betrachtet (Komma links)
Dividiere invers	DVI	n	$\langle A \rangle := \langle n \rangle : \langle A \rangle$	+	x	$1; \langle n \rangle : \langle A \rangle$	1; Rest · 2 ⁴⁸	1; +0		+0	x	**	≠1	R	222	62	
Dividiere doppelt lang	DVD	n	$\langle A \rangle := \langle A, Q \rangle : \langle n \rangle$	+	x	$1; \langle A, Q \rangle : \langle n \rangle$	1; Rest · 2 ⁴⁸	1; +0		+0	x	**	≠1	R	228	61	

Addiere in AQ	AQ	n	$\langle A, Q \rangle := \langle A, Q \rangle + \langle n \rangle$	+	x	$1, 1; \langle A, Q \rangle + \langle n \rangle$	$t_n; \langle n \rangle$			x	≥	≠1		R	17	7E	Operand und Ergebnis: $\langle A \rangle_v$ kann ≠ $\langle Q \rangle_v$
Subtrahiere in AQ	SBQ	n	$\langle A, Q \rangle := \langle A, Q \rangle - \langle n \rangle$	+	x	$1, 1; \langle A, Q \rangle - \langle n \rangle$	$t_n; \langle n \rangle$			x	≥	≠1		R	17	7F	
Multipliziere	ML	n	$\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle$	+	x	$1, 1; \langle A \rangle \cdot \langle n \rangle$	$t_n; \langle n \rangle$			+0	x	≠1		R	55	54	Ergebnis: $\langle A \rangle_v = \langle Q \rangle_v$
Multipliziere negativ	MLN	n	$\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle$	+	x	$1, 1; -\langle A \rangle \cdot \langle n \rangle$	$t_n; \langle n \rangle$			+0	x	≠1		R	55	58	
Multipliziere akkumulierend	MLA	n	$\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	+	x	$1, 1; \langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	$t_n; \langle n \rangle$			+0	x	≥	≠1	R	66	56	$\langle H \rangle_v$ und $\langle Q \rangle_v$ ohne Bedeutung für die Ausführung des Befehls Operand: $\langle H \rangle_v$ kann ≠ $\langle Q \rangle_v$ Ergeb.: $\langle A \rangle_v$ kann ≠ $\langle Q \rangle_v$
Multipliziere akkumulierend negativ	MAN	n	$\langle A, Q \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	+	x	$1, 1; -\langle A \rangle \cdot \langle n \rangle + \langle H, Q \rangle$	$t_n; \langle n \rangle$			+0	x	≥	≠1	R	66	5A	

Addiere Adressenteil	AA	z	$\langle A \rangle := \langle A \rangle + z$	+		$TK = 0: 0; \langle A \rangle \cdot 16^{\pm z}$ $TK = 1: 1; \langle A \rangle \cdot v, z$	$1; 0, z$ $1; v, z$							R	13	98	nach Modifiz. $TK = 0: z < 2^7$ $TK = 1: z < 2^{23}$ $TK = 2: z < 2^{16}$ $TK = 3: z < 2^{23}$
Subtrahiere Adressenteil	SBA	z	$\langle A \rangle := \langle A \rangle - z$	+		$TK = 2: 2; \langle A \rangle - 32; (\langle A \rangle_{32-48} \pm z)$ $TK = 3: 3; \langle A \rangle \pm v, z$	$1; 0, z$ $1; v, z$							R	13	99	

* gerundet ** Voraussetzung: bei DV, DVD: $|\langle A \rangle| < |\langle n \rangle|$
bei DVI : $|\langle n \rangle| < |\langle A \rangle|$ im anderen Fall: Gleitkommaergebnis, BÜ-Alarm

Gleitkomma-Arithmetik

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle A \rangle$	$\langle n \rangle$	$\langle D \rangle$	$\langle Q \rangle$	$\langle Y \rangle$	$\langle M \rangle$	BU	TK	Werk	Takte	Int.	Bemerkungen
Gleitkomma <u>a</u> ddiere	GA	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle$	+	x	$0; \langle A \rangle + \langle n \rangle^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	28	4B	
Gleitkomma <u>a</u> ddiere <u>B</u> etrag	GAB	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle $	+	x	$0; \langle A \rangle + \langle n \rangle ^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	28	52	
Gleitkomma <u>a</u> ddiere im <u>S</u> peicher	GAC	n	$\langle n \rangle := \langle A \rangle + \langle n \rangle$	+			$0; \langle A \rangle + \langle n \rangle^*$	$0; \langle A \rangle + \langle n \rangle^*$	$0; +0$	**	x	>	$\neq 0$	B	R	37	4A
Gleitkomma <u>s</u> ubtrahiere	GSB	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle$	+	x	$0; \langle A \rangle - \langle n \rangle^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	28	4F	
Gleitkomma <u>s</u> ubtrahiere <u>B</u> etrag	GSBB	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle $	+	x	$0; \langle A \rangle - \langle n \rangle ^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	28	53	
Gleitkomma <u>s</u> ubtrahiere im <u>S</u> peicher	GSBC	n	$\langle n \rangle := \langle n \rangle - \langle A \rangle$	+			$0; \langle n \rangle - \langle A \rangle^*$	$0; \langle n \rangle - \langle A \rangle^*$	$0; +0$	**	x	>	$\neq 0$	B	R	37	4E
Gleitkomma <u>s</u> ubtrahiere <u>i</u> nv <u>e</u> rs	GSBI	n	$\langle A \rangle := \langle n \rangle - \langle A \rangle$	+	x	$0; \langle n \rangle - \langle A \rangle^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	28	48	
Gleitkomma <u>s</u> ubtrahiere von <u>D</u>	GSBD	n	$\langle A \rangle := \langle D \rangle - \langle n \rangle$	+	x	$0; \langle D \rangle - \langle n \rangle^*$			$0; +0$	**	x	>	$\neq 0$	R	30	4C	
Gleitkomma <u>m</u> ultipl <u>i</u> ziere	GML	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle$	+	x	$0; \langle A \rangle \cdot \langle n \rangle^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	54	5E	
Gleitkomma <u>m</u> ultipl <u>i</u> ziere <u>n</u> egativ	GMLN	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle$	+	x	$0; -\langle A \rangle \cdot \langle n \rangle^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	54	5C	
Gleitkomma <u>m</u> ultipl <u>i</u> ziere <u>a</u> kkumulierend	GMLA	n	$\langle A \rangle := \langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	x	$0; \langle A \rangle \cdot \langle n \rangle + \langle H \rangle^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	97	5F	
Gleitkomma <u>m</u> ultipl <u>i</u> ziere <u>a</u> kk <u>u</u> m. <u>n</u> egativ	GMAN	n	$\langle A \rangle := -\langle A \rangle \cdot \langle n \rangle + \langle H \rangle$	+	x	$0; -\langle A \rangle \cdot \langle n \rangle + \langle H \rangle^*$		$t_n; \langle n \rangle$	$0; +0$	**	x	>	$\neq 0$	R	97	5D	
Gleitkomma <u>d</u> i <u>v</u> id <u>i</u> ere	GDV	n	$\langle A \rangle := \langle A \rangle : \langle n \rangle$	+	x	$0; \langle A \rangle : \langle n \rangle^*$		$0; +0$	$0; +0$	**	x	>	$\neq 0$	R	213	64	
Gleitkomma <u>d</u> i <u>v</u> id <u>i</u> ere <u>i</u> nv <u>e</u> rs	GDVI	n	$\langle A \rangle := \langle n \rangle : \langle A \rangle$	+	x	$0; \langle n \rangle : \langle A \rangle^*$		$0; +0$	$0; +0$	**	x	>	$\neq 0$	R	216	66	
<u>A</u> ddiere <u>u</u> nnormalisiert	AU	n	$\langle A \rangle := \langle A \rangle + \langle n \rangle$	+	x	$0; \langle A \rangle + \langle n \rangle$		$t_n; \langle n \rangle$	$0; +0$	+0	x	>	$\neq 0$	R	26	49	
<u>S</u> ubtrahiere <u>u</u> nnormalisiert	SBU	n	$\langle A \rangle := \langle A \rangle - \langle n \rangle$	+	x	$0; \langle A \rangle - \langle n \rangle$		$t_n; \langle n \rangle$	$0; +0$	+0	x	>	$\neq 0$	R	26	4D	
Bilde <u>r</u> eziproken Wert	REZ	n	$\langle A \rangle := 1 : \langle n \rangle$	+	x	$0; 1 : \langle n \rangle^*$		$0; +0$	$0; +0$	**	x	>	$\neq 0$	R	213	65	

			$\langle A, Q \rangle$			$\langle H \rangle$											
Doppelte Genauigkeit: <u>A</u> ddiere	DA	n	$\langle A, Q \rangle := \langle A, Q \rangle + \langle n, n+2 \rangle$			$0, 1; \langle A, Q \rangle + \langle n, n+2 \rangle^*$		$1; +0$	$1; +0$	**	x	>	$2, 3$	BR	106	F0	
Doppelte Genauigkeit: <u>S</u> ubtrahiere	DSB	n	$\langle A, Q \rangle := \langle A, Q \rangle - \langle n, n+2 \rangle$			$0, 1; \langle A, Q \rangle - \langle n, n+2 \rangle^*$		$1; +0$	$1; +0$	**	x	>	$2, 3$	BR	106	F1	
Doppelte Genauigkeit: <u>M</u> ultipl <u>i</u> ziere	DML	n	$\langle A, Q \rangle := \langle A, Q \rangle \cdot \langle n, n+2 \rangle$			$0, 1; \langle A, Q \rangle \cdot \langle n, n+2 \rangle^*$		$1; +0$	$1; +0$	**	x	>	$2, 3$	BR	243	F2	
Gleitkomma <u>m</u> ultipl <u>i</u> z. auf <u>d</u> oppelte Genauigk.	MLD	n	$\langle A, Q \rangle := \langle A \rangle \cdot \langle n \rangle$		x	$0, 1; \langle A \rangle \cdot \langle n \rangle^*$		$1; +0$	$1; +0$	**	x	>	$\neq 0$	BR	64	F3	

$\langle A \rangle_t = \langle n \rangle_t = 0$
 $\langle Q \rangle_t = \langle n+2 \rangle_t = 1$

			$\langle A \rangle$														
<u>A</u> ddiere <u>A</u> dressenteil (TK = 0)	AA	z	$\langle A \rangle := \langle A \rangle \cdot 16^z$	+		$0; \langle A \rangle \cdot 16^{+z}$		$1; 0, z$				>		R	13	98	
<u>S</u> ubtrahiere <u>A</u> dressenteil (TK = 0)	SBA	z	$\langle A \rangle := \langle A \rangle \cdot 16^{-z}$	+		$0; \langle A \rangle \cdot 16^{-z}$		$1; 0, z$				>		R	13	99	

nur bei $\langle A \rangle_t = 0$; bei TK $\neq 0$
 siehe Festkomma-Arithmetik.
 $|z| < 2^7$ nach Modifizierung

* normalisiert und gerundet

** Anzahl der Binärstellen um die das Ergebnis normalisiert wurde.

Falls Ergebnis = +0 oder Exponentenunterlauf: $\langle Y \rangle := +0$

Boolesche Operationen

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle A \rangle$	$\langle D \rangle$	$\langle M \rangle$	Alarm	BÜ	Werk	Take	Int.	Bemerkungen
<u>VEL</u>	VEL	n	$\langle A \rangle := \langle A \rangle \vee \langle n \rangle$	+	x	$t_{n,x}; \langle A \rangle \vee \langle n \rangle$	$t_n; \langle n \rangle$		x		R	5	68	
<u>AUT</u>	AUT	n	$\langle A \rangle := \langle A \rangle \neq \langle n \rangle$	+	x	$t_{n,x}; \langle A \rangle \neq \langle n \rangle$	$t_n; \langle n \rangle$		x		R	5	69	
<u>ET</u>	ET	n	$\langle A \rangle := \langle A \rangle \wedge \langle n \rangle$	+	x	$t_{n,x}; \langle A \rangle \wedge \langle n \rangle$	$t_n; \langle n \rangle$		x		R	5	6A	
Setze <u>zusammen</u>	ZUS	n	$\langle A \rangle_x := \langle A \rangle_x$: für $\langle H \rangle_x = 0$ $\langle A \rangle_x := \langle n \rangle_x$: für $\langle H \rangle_x = L$	+	x	$t_{n,x}(\langle A \rangle_x, \langle n \rangle_x) \dots$; (s. Wirkung)	$t_n; \langle n \rangle$		x		R	5	6B	
<u>VEL</u> Adressenteil	VLA	z	$\langle A \rangle := \langle H \rangle \vee z$	+		$t_H; \langle H \rangle \vee (0, z)$	$1; 0, z$				R	8	88	} z: 0...65535
<u>AUT</u> Adressenteil	ATA	z	$\langle A \rangle := \langle H \rangle \neq z$	+		$t_H; \langle H \rangle \neq (0, z)$	$1; 0, z$				R	8	89	
<u>ET</u> Adressenteil	ETA	z	$\langle A \rangle := \langle H \rangle \wedge z$	+		$t_H; \langle H \rangle \wedge (0, z)$	$1; 0, z$				R	8	8A	

Halbwort-Arithmetik

			$\langle A \rangle$			$\langle Q \rangle$	$\langle D \rangle$	$\langle Y \rangle$							
Addiere <u>Halbwort</u>	A2	m	$\langle A \rangle := \langle A \rangle_{25-48} + \langle m \rangle$	+	x	$t_A; v, (\langle A \rangle_{25-48} + \langle m \rangle)$		$t_A; +0$				\geq	R	14	7C
Subtrahiere <u>Halbwort</u>	SB2	m	$\langle A \rangle := \langle A \rangle_{25-48} - \langle m \rangle$	+	x	$t_A; v, (\langle A \rangle_{25-48} - \langle m \rangle)$		$t_A; +0$				\geq	R	14	7D
Multipliziere <u>Halbwort</u>	M2	m	$\langle A \rangle := \langle A \rangle \cdot \langle m \rangle$	+	x	$t_A; \langle A \rangle \cdot \langle m \rangle$	$t_A; +0$	$t_A; +0$	+0			\geq	R	49,5	7A
Multipliziere <u>Halbwort</u> negativ	M2N	m	$\langle A \rangle := -\langle A \rangle \cdot \langle m \rangle$	+	x	$t_A; -\langle A \rangle \cdot \langle m \rangle$	$t_A; +0$	$t_A; +0$	+0			\geq	R	49,5	78
Multipliziere <u>Halbwort</u> mit <u>Rundung</u>	M2R	m	$\langle A \rangle := \langle A \rangle \cdot \langle m \rangle$ gerundet	+	x	$t_A; (\langle A \rangle \cdot \langle m \rangle)$ gerundet	$t_A; +0$	$t_A; +0$	+0				R	36,5	7B
Multipliziere <u>Halbwort</u> negativ mit <u>Rundung</u>	M2NR	m	$\langle A \rangle := -\langle A \rangle \cdot \langle m \rangle$ gerundet	+	x	$t_A; (-\langle A \rangle \cdot \langle m \rangle)$ gerundet	$t_A; +0$	$t_A; +0$	+0				R	36,5	79

Teilwort-Arithmetik

$\langle A \rangle$						$\langle D \rangle$						
<u>Addiere Teilwort</u>	AT	n	$\langle A \rangle := \langle A \rangle_x + \langle n \rangle_x$	+	x	Hilfsgröße qr := um p Stellen rechts im Kreis geschifteter $\langle Q \rangle$ Hilfsgröße nr := um p Stellen rechts geschifteter $\langle n \rangle$	$t_q; \langle Q \rangle$	$>$	R	21+	F4	p = Anzahl der rechts in Q anstehenden L-Bits Falls $\langle Q \rangle = LL...L$ dann p = 0
<u>Subtrahiere Teilwort</u>	SBT	n	$\langle A \rangle := \langle A \rangle_x - \langle n \rangle_x$	+	x	Das Nullfeld von qr schneidet in nr und $\langle A \rangle$ Teilwörter aus; diese werden addiert bzw. subtrahiert. Die anderen Stellen werden auf 0 gelöscht.	$t_q; \langle Q \rangle$	$<$	R	2p	F5	Die Zahlen werden als positive, vorzeichenlose, ganze Festkommazahlen aufgefaßt.
						t_A ; Ergebnis						

Index-Arithmetik

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle B \rangle$	$\langle i \rangle$	$\langle i_R \rangle$	Werk	Takte	Int.	Bemerkungen
Erhöhe <u>B</u> um <u>Speicher</u>	HBC	m	$\langle B \rangle := \langle B \rangle + \langle m \rangle$		x	$\langle B \rangle + \langle m \rangle$			B	6,5	3C	
Vermindere <u>B</u> um <u>Speicher</u>	VBC	m	$\langle B \rangle := \langle B \rangle - \langle m \rangle$		x	$\langle B \rangle - \langle m \rangle$			B	8,5	15	
Erhöhe <u>B</u> um <u>Adressenteil</u>	HBA	z	$\langle B \rangle := \langle B \rangle + z$			$\langle B \rangle + z$			B	7,5	11	z: 0...65 535
Vermindere <u>B</u> um <u>Adressenteil</u>	VBA	z	$\langle B \rangle := \langle B \rangle - z$			$\langle B \rangle - z$			B	8,5	13	
Erhöhe <u>B</u> um <u>Parameter</u> mal <u>Indexzelle</u>	HBPX	p i	$\langle B \rangle := \langle B \rangle + p \cdot \langle i \rangle$			$\langle B \rangle + p \cdot \langle i \rangle$			B	5,5 p +3	0F	p: $\pm 1 \dots \pm 15$
Erhöhe <u>Indexzelle</u> um <u>Parameter</u>	HXP	p i	$\langle B \rangle := \langle i \rangle + p$ $\langle i \rangle := \langle i \rangle + p$			$\langle i \rangle + p$	$\langle i \rangle + p$		B	10,5	2C	p: $\pm 0 \dots \pm 127$
Erhöhe <u>Indexzelle</u> um <u>Indexzelle</u>	HXX	i _L i _R	$\langle B \rangle := \langle i_R \rangle + \langle i_L \rangle$ $\langle i_R \rangle := \langle i_R \rangle + \langle i_L \rangle$			$\langle i_R \rangle + \langle i_L \rangle$		$\langle i_R \rangle + \langle i_L \rangle$	B	14,5	2E	
Vermindere <u>Indexzelle</u> um <u>Indexzelle</u>	VXX	i _L i _R	$\langle B \rangle := \langle i_R \rangle - \langle i_L \rangle$ $\langle i_R \rangle := \langle i_R \rangle - \langle i_L \rangle$			$\langle i_R \rangle - \langle i_L \rangle$		$\langle i_R \rangle - \langle i_L \rangle$	B	14,5	2F	
<u>Register</u> und <u>Indexzelle</u>	RX	s i	$\langle B \rangle := \langle i \rangle \pm \langle s_1 \rangle$ Falls $s_3 = C$: $\langle i \rangle := \langle i \rangle \pm \langle s_1 \rangle$			$\langle i \rangle \pm \langle s_1 \rangle$ $\langle i \rangle \pm \langle s_1 \rangle$	 nur falls $s_3 = C$ $\langle i \rangle \pm \langle s_1 \rangle$		B R	12	* 8D	s_1 : A, Q, D, H (rechtes Halbwort) oder B s_2 : leer = positiv + N = negativ - } statt \pm s_3 : leer = nicht zurückspeich. C = zurückspeichern

*siehe Internspezifikation auf Seite 23

Setzen und Löschen

			$\langle A \rangle$		$\langle H \rangle$	$\langle s_2 \rangle$							
<u>Bringe</u> <u>Adressenteil</u>	BA	z	$\langle A \rangle := z$	+		1;v,z				R	2	8E	z: 0...65535
<u>Bringe</u> <u>Adressenteil</u> und <u>reserviere</u>	BAR	z	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := z$	+		1;v,z	$t_A; \langle A \rangle$			R	3	DC	
<u>Bringe</u> <u>Adressenteil</u> <u>negativ</u>	BAN	z	$\langle A \rangle := -z$	+		1;-(v,z)				R	5	DF	
<u>Bringe</u> <u>Adressenteil</u> <u>negativ</u> und <u>reserviere</u>	BANR	z	$\langle H \rangle := \langle A \rangle$ $\langle A \rangle := -z$	+		1;-(v,z)	$t_A; \langle A \rangle$			R	5	DD	
<u>Lösche</u> <u>Register</u>	LR	s	$\langle s_2 \rangle := +0$ $\langle s_2 \rangle_t := s_1$	+				$s_1; +0$		R	3	9A	s_1 : 0,1,2 oder 3 (TK) s_2 : A,Q,D und H
<u>Lösche</u> in <u>A</u>	LA	s	$\langle A \rangle_s := 0$ $\langle M \rangle := 0$ nur bei $s = M$	+	W. r. r. r. r. $s = F: \langle A \rangle_{1-40} := 0 \hat{=}$ Mantissenteil $s = 2: \langle A \rangle_{1-24} := 0 \hat{=}$ linkes Halbwort $s = E: \langle A \rangle_{41-48} := 0 \hat{=}$ Exponententeil $s = 3: \langle A \rangle_{33-40} := 0 \hat{=}$ rechtes Drittel $s = V: \langle A \rangle_{1-2} := 0 \hat{=}$ Vorzeichenstellen $s = M: \langle M \rangle := 0 \hat{=}$ Markenregister				Die Spezifikationen F bis M können kombiniert verwendet werden	R	4	8E	
					$s = H: \langle A \rangle_{1-42} := 0 \hat{=}$ ohne rechte Hexade $s = T: \langle A \rangle_{1-44} := 0 \hat{=}$ ohne rechte Tetrade				H und T nur einzeln od. mit M erlaubt				

Bezeichnung	Code	adr	Wirkung	mod2	R		<i>	<U>	Werk	Take	Int.	Bemerkungen
Index: <u>B</u> ringe <u>A</u> dressenteil	XBA	z	 := z			z			B	1	01	z: 0...65535
Index: <u>B</u> ringe <u>A</u> dressenteil <u>n</u> egativ	XBAN	z	 := -z			-z			B	2	19	
Setze <u>I</u> ndex	ZX	p i	<i> := p := p			v,p	v,p		B	6	1A	p: ±0...±127
Setze <u>U</u> nterprogrammregister	ZU	i	<U> := i	+				i	B	1	3E	

Setze <u>T</u> ypenkennung im <u>R</u> egister	ZTR	s	<s ₂ > _t := s ₁ <M> := L, falls s ₃ = M	+		s ₁	L, falls s ₃ = M		R	5	92	s ₁ : 0,1,2,3 oder leer s ₂ : A,Q,D,H oder leer s ₃ : leer oder M
Setze <u>T</u> ypenkennung <u>0</u>	ZT0	n	<n> _t := 0	+		0			B	13	C8	
Setze <u>T</u> ypenkennung <u>1</u>	ZT1	n	<n> _t := 1	+		1			B	13	C9	
Setze <u>T</u> ypenkennung <u>2</u>	ZT2	n	<n> _t := 2	+		2			B	13	CA	
Setze <u>T</u> ypenkennung <u>3</u>	ZT3	n	<n> _t := 3	+		3			B	13	CB	

<u>L</u> ösche <u>S</u> peicher	LC	n	<n> := +0 <n> _n := <n> _n	+		t _n ; +0	<n> _n				B	13	33	
<u>L</u> ösche <u>m</u> arkiert	LMT	n	<n> := +0 <n> _n := L	+		t _n ; +0	L			2,3	B	13	32	
<u>L</u> ösche <u>M</u> arke im <u>S</u> peicher	LMC	n	<n> _n := 0	+			0			2,3	B	13	31	
Setze <u>M</u> arke im <u>S</u> peicher	ZMC	n	<n> _n := L	+			L			2,3	B	13	30	
<u>L</u> ösche und setze <u>M</u> erklichter	LZL	s _L s _R	<K> _{s_R} := 0 <K> _{s_L} := L				L	0			B	3	10	s: Merklichter 0,1,2,3,4,5,6,7 und 8 0 bedeutet kein Merklicht (0 muß angegeben werden)
<u>N</u> egiere <u>M</u> erklichter	NL	s	<K> _s := <K> _s invertiert					invertiert			B	3	12	
Setze <u>I</u> ndexbasis	ZI	m	<X> := <m> ₃₋₂₄						<m> ₃₋₂₄		B	17 x + 58	B7	x: Anzahl der Indexregister, die zurückgespeichert werden

Sprünge

Bezeichnung	Code	adr	Wirkung	mod2	R	$\langle F \rangle_{9-24}$	$\langle F \rangle_{1-24}$	$\langle H \rangle$	$\langle B \rangle$	$\langle U \rangle$	U-Alarm	Werk	Takte	Int.	Bemerkungen
<u>Null</u> befehl	NULL	z	keine Wirkung		x							B	1	00	Nur übliche Erhöhung des Befehlsfolgeregisters. z: ohne Bedeutung, muß angegeben werden.
<u>Warte</u> befehl	WB	z	z Uhrimpulse warten (Uhrimpulse alle 10µs)									B R	s.Be-merkung.	F8	z: 0...65535 Takte: bei z = 0: 3 Takte bei z ≠ 0: (10z-5)µs
<u>Spr</u> inge	S	m	$\langle F \rangle_{9-24} := m$	+		m						B	1	36	
<u>Spr</u> inge nach <u>Er</u> setzung	SE	m	$\langle F \rangle := \langle m \rangle + \text{mod2}$	sp	x		$\langle m \rangle + \text{mod2}$					B	2	BC	Sprung in and. Großseite mögl.
<u>Spr</u> inge und bringe $\langle F \rangle + 1$ nach <u>B</u>	SFB	m	$\langle B \rangle := \langle F \rangle + 1$ $\langle F \rangle_{9-24} := m$	+		m			$\langle F \rangle + 1$			B	1	3A	
<u>Spr</u> inge nach <u>Er</u> setzung und bringe $\langle F \rangle + 1$ nach <u>B</u>	SFBE	m	$\langle B \rangle := \langle F \rangle + 1$ $\langle F \rangle_{1-24} := \langle m \rangle + \text{mod2}$	sp			$\langle m \rangle + \text{mod2}$		$\langle F \rangle + 1$			B R	15	FA	Sprung in andere Großseite mögl.
<u>Spr</u> inge in <u>U</u> nterprogramm	SU	m	$\langle U \rangle := \langle U \rangle + 1$ $\langle U \rangle := \langle F \rangle + 1$ $\langle F \rangle_{9-24} := m$	+		m				$\langle U \rangle + 1$	$\langle U \rangle_{alt} = 254$	B	8	38	$\langle U \rangle$: 0...255 ($\langle U \rangle$: 255 + 1 = 0)
<u>Spr</u> inge in <u>U</u> nterprogramm nach <u>Er</u> setzung	SUE	m	$\langle U \rangle := \langle U \rangle + 1$ $\langle U \rangle := \langle F \rangle + 1$ $\langle F \rangle := \langle m \rangle + \text{mod2}$	sp	x		$\langle m \rangle + \text{mod2}$			$\langle U \rangle + 1$	$\langle U \rangle_{alt} = 254$	B	8	BD	$\langle U \rangle$: 0...255 ($\langle U \rangle$: 255 + 1 = 0) Sprung in and. Großseite mögl.
<u>Spr</u> inge ins <u>S</u> ystem und <u>r</u> eserviere	SSR	$P_L P_R$	$\langle F \rangle := \langle a + 6 \rangle$ $\langle B \rangle := \text{adr}$	+			$\langle a + 6 \rangle$ Sprung in den Abwickler zur Erbringung von Dienstleistungen gem. P_L und P_R ; s. spezielle Beschreibungen		adr			B R	48	BB	$\left. \begin{matrix} P_L \\ P_R \end{matrix} \right\} : 0...255$ $a = \langle BL \rangle \cdot 2^8 = \text{Anfangsadresse des Leitblocks}$

Sprungbedingung							$\langle A \rangle$									
<u>Spr</u> inge wenn <u>i</u> dentisch <u>0</u>	SIO	m	$\langle A \rangle = 0$	+		m *							B R	1	A4	
<u>Spr</u> inge wenn <u>n</u> icht <u>0</u>	SNO	m	$\langle A \rangle \neq 0$	+		m *							B R	1	A7	
<u>Spr</u> inge wenn <u>g</u> rößer <u>g</u> leich <u>0</u>	SGGO	m	$\langle A \rangle \geq 0$	+		m *							B R	1	A6	$\langle A \rangle_t = 2$ oder 3: Sprung immer!
<u>Spr</u> inge wenn <u>g</u> rößer <u>0</u>	SGO	m	$\langle A \rangle > 0$	+		m *							B R	1	D4	
<u>Spr</u> inge wenn <u>k</u> leiner <u>g</u> leich <u>0</u>	SKGO	m	$\langle A \rangle \leq 0$	+		m *							B R	1	A5	
<u>Spr</u> inge wenn <u>k</u> leiner <u>0</u>	SKO	m	$\langle A \rangle < 0$	+		m *							B R	1	D5	$\langle A \rangle_t = 2$ oder 3: Sprung nie!
<u>Spr</u> inge wenn <u>r</u> echtes Bit in A gesetzt	SR	m	$\langle A \rangle_{48} = L$	+		m *							B R	1	B8	
<u>Spr</u> inge wenn <u>r</u> echtes Bit in A <u>n</u> icht gesetzt	SRN	m	$\langle A \rangle_{48} = 0$	+		m *							B R	1	BA	
<u>Spr</u> inge wenn <u>i</u> dentisch	SI	m	$\langle A \rangle = \langle H \rangle$	+		m *	norm.**	norm.**					B R	** 6	AC	
<u>Spr</u> inge wenn <u>n</u> icht identisch	SN	m	$\langle A \rangle \neq \langle H \rangle$	+		m *	norm.**	norm.**					B R	** 6	AD	
<u>Spr</u> inge wenn <u>g</u> rößer <u>g</u> leich	SCG	m	$\langle A \rangle \geq \langle H \rangle$	+		m *	norm.**	norm.**					B R	** 6	AF	
<u>Spr</u> inge wenn <u>g</u> rößer	SG	m	$\langle A \rangle > \langle H \rangle$	+		m *	norm.**	norm.**					B R	** 6	AB	
<u>Spr</u> inge wenn <u>k</u> leiner <u>g</u> leich	SKG	m	$\langle A \rangle \leq \langle H \rangle$	+		m *	norm.**	norm.**					B R	** 6	AE	
<u>Spr</u> inge wenn <u>k</u> leiner	SK	m	$\langle A \rangle < \langle H \rangle$	+		m *	norm.**	norm.**					B R	** 6	AA	

Bezeichnung	Code	adr	Sprungbedingung	mod? R	$\langle F \rangle_{0-24}$	$\langle i \rangle$	$\langle M \rangle \langle K \rangle_s$	$\langle D \rangle$	$\langle U \rangle$	Alarm	Werk	Jakte	Int.	Bemerkungen					
Springe wenn Index <u>identisch</u> 0	SXI	m	$\langle B \rangle = \pm 0$		m *						B	1	1	24	$\langle B \rangle_1 = \langle B \rangle_v$				
Springe wenn Index <u>nicht</u> identisch 0	SXN	m	$\langle B \rangle \neq \pm 0$		m *						B	1	1	27					
Springe wenn Index <u>größer</u> <u>gleich</u> 0	SXGG	m	$\langle B \rangle \geq \pm 0$		m *						B	1	1	25					
Springe wenn Index <u>größer</u> 0	SXG	m	$\langle B \rangle > \pm 0$		m *						B	1	1	CE					
Springe wenn Index <u>kleiner</u> <u>gleich</u> 0	SXKG	m	$\langle B \rangle \leq \pm 0$		m *						B	1	1	26					
Springe wenn Index <u>kleiner</u> 0	SXK	m	$\langle B \rangle < \pm 0$		m *						B	1	1	CF					
Springe wenn Indexgröße <u>rechtes</u> Bit = L	SXR	m	$\langle B \rangle_{24} = L$		m *						B	1	1	B2					
Springe wenn Indexgröße <u>rechtes</u> Bit <u>nicht</u> L	SXRN	m	$\langle B \rangle_{24} = 0$		m *						B	1	1	B3					
Springe und <u>zähle</u> wenn Index <u>kleiner</u> 0	SZX	p i	$\langle i \rangle < \pm 0$		$\langle F \rangle + p *$	$\langle i \rangle + 1 *$					B	11	2	0A	$\langle i \rangle_1 = \langle i \rangle_v$ p: $\pm 0 \dots \pm 127$				
Springe wenn <u>Marke</u>	SM	m	$\langle M \rangle = L$	+	m *		0				B	1	1	34					
Springe wenn <u>Marke</u> <u>nicht</u>	SMN	m	$\langle M \rangle = 0$	+	m *		0				B	1	1	35					
Springe wenn <u>arithmetischer</u> Alarm (BÜ-Alarm)	SAA	m	BÜ-Alarm	+	m *					BÜ-Alarm wird gelöscht	B R	1	1	A9					
Springe wenn <u>Alarm</u> (Typenkennung)	SAT	m	TK-Alarm	+	m *					TK-Alarm wird gelöscht	B R	1	1	A8					
Springe wenn <u>Typenkennung</u>	ST	p s	$\langle s_2 \rangle_t = s_1$	+	$\langle F \rangle + p *$						B R	10	3	90	p: $\pm 0 \dots \pm 127$ s_1 : 0,1,2 oder 3 (TK) s_2 : A,Q,D oder H				
Springe wenn <u>Typenkennung</u> <u>nicht</u>	STN	p s	$\langle s_2 \rangle_t \neq s_1$	+	$\langle F \rangle + p *$						B R	10	3	91					
Springe wenn <u>Bit</u> gesetzt	SBIT	p s	$\langle s_2 \rangle_{s_1} = L$		$\langle F \rangle + p *$	Tabelle für Zeitberechnung: s_1 1 2 ... 14 15 16 17 ... 31 32 33 ... 47 48 q 16 15 ... 3 2 1 16 ... 2 1 16 ... 2 1					B R	7+2q	F9	s_1 : Bitnummer 1...48 s_2 : Register A, Q, D oder H p : Sprungweite $\pm 0 \dots \pm 127$ q : s. Tab. für Zeitberechn.					
Springe wenn <u>Exponent</u> <u>größer</u> <u>gleich</u>	SEGG	p _L p _R	$\langle A \rangle_{41-48} \geq p_R$	+	$\langle F \rangle + p_L *$			$t_A; \langle A \rangle$		falls $\langle A \rangle_t \neq 0$: TK-Alarm	B R	10	4	93	p _L : Sprungweite p: $\pm 0 \dots \pm 127$ p _R : Vergl.-Exponent ****				
Springe wenn <u>Merklicht</u>	SL	p s	eines der $\langle K \rangle_s = L$		$\langle F \rangle + p *$						B	9	1	1E	p: $\pm 0 \dots \pm 127$ s: Merklichter 1,2,... und 8				
Springe wenn <u>Merklicht</u> und <u>lösche</u>	SLL	p s	eines der $\langle K \rangle_s = L$		$\langle F \rangle + p *$		0				B	9	1	1F					
Springe wenn <u>Merklicht</u> <u>nicht</u>	SLN	p s	alle $\langle K \rangle_s = 0$		$\langle F \rangle + p *$						B	9	1	1C					
Springe wenn <u>Merklicht</u> <u>nicht</u> sonst <u>lösche</u>	SNL	p s	alle $\langle K \rangle_s = 0$		$\langle F \rangle + p *$		0				B	9	1	1D					
															$\langle A \rangle$	$\langle Y \rangle$			
Prüfe <u>Dreierprobe</u> und springe wenn richtig	PDP	n	DP = richtig	+	$\langle F \rangle + 2 *$		$t_n; \langle n \rangle$		$\langle n \rangle_{5-8} 0L, DP, \langle n \rangle_t$		R	21	21	B9	DP: Dreierprobenbits Transp. unverän. bei jeder TK DP <u>nicht</u> richt.: kein DP-Alarm				

* bei erfüllter Sprungbedingung
 ** wenn $\langle A \rangle_t = \langle H \rangle_t = 0$: $\langle A \rangle$:= $\langle A \rangle$ normalisiert
 $\langle H \rangle$:= $\langle H \rangle$ normalisiert
 ist der Exponent +0 wird er zu -0, zur
 Ausführungszeit kommt ein Takt hinzu

*** bei nicht erfüllter Sprungbedingung
 **** 0...127 (positiv), NO...N127 (negativ)

Modifizieren

Bezeichnung	Code	adr	Wirkung	mod2	R	mod1	mod2 ₁	(B)	(i)	Werk	lakte	Int.	Bemerkungen
Modifiziere in jedem Fall	MF	i	mod1 := <i> + mod2 := <i> + mod2	sp		<i> *	0	<i> *		B	1	0B	
Modifiziere in jedem Fall mit unveränd. B	MFU	i	mod1 := <i> + mod2	sp		<i> *	0			B	1	08	Bits 9 - 11 = 00L
Modifiziere aus Speicher in jedem Fall	MCF	m	mod1 := <m> + mod2 := <m> + mod2	sp	x	<m> *	0	<m> *		B	2	16	
Modifiziere aus Speicher in jedem Fall mit unverändertem B	MCFU	m	mod1 := <m> + mod2	sp	x	<m> *	0			B	1	3D	
Modifiziere doppelt	MD	i _L i _R	mod1 := <i _{Rmod2 := <i_{L := <i_{L<td>sp</td><td></td><td><i_{R<td><i_{L<td><i_{L<td></td><td>B</td><td>5</td><td>09</td><td></td>}</td>}</td>}</td>}}}	sp		<i _{R<td><i_{L<td><i_{L<td></td><td>B</td><td>5</td><td>09</td><td></td>}</td>}</td>}	<i _{L<td><i_{L<td></td><td>B</td><td>5</td><td>09</td><td></td>}</td>}	<i _{L<td></td><td>B</td><td>5</td><td>09</td><td></td>}		B	5	09	
Modifiziere	M	i	 := <i> + mod2 mod2 := <i> + mod2	sp		<i> *	<i> *	<i> *		B	1	08	Bits 9 - 11 = L00
Modifiziere nach Erhöhung	MH	p i	mod2 := <i> + p := <i> + p <i> := <i> + p				<i> + p	<i> + p	<i> + p	B	10,5	2D	p: ±0...±127
Modifiziere nach Erhöhung um Indexzelle	MHX	i _L i _R	mod2 := <i _{LR := <i_{LR<i_{RLR<td></td><td></td><td></td><td><i_{LR<td><i_{LR<td><i_{RLR<td>B</td><td>14,5</td><td>0E</td><td></td>}</td>}</td>}</td>}}}				<i _{LR<td><i_{LR<td><i_{RLR<td>B</td><td>14,5</td><td>0E</td><td></td>}</td>}</td>}	<i _{LR<td><i_{RLR<td>B</td><td>14,5</td><td>0E</td><td></td>}</td>}	<i _{RLR<td>B</td><td>14,5</td><td>0E</td><td></td>}	B	14,5	0E	
Modifiziere mit Register und Indexzelle	MRX	s i	mod2 := <i> ± <s _{1 := <i> ± <s_{1 falls s₃ = C: <i> := <i> ± <s_{1<td></td><td></td><td></td><td><i> ± <s_{1<td><i> ± <s_{1<td>falls s₃ = C: <i> ± <s_{1<td>B</td><td>12</td><td>8D **</td><td>s₁: A, Q, D, H (rechtes Halbwort) oder B s₂: leer: + } statt ± : N : - s₃: leer: nicht speichern : C : zurückspeichern</td>}</td>}</td>}</td>}}}				<i> ± <s _{1<td><i> ± <s_{1<td>falls s₃ = C: <i> ± <s_{1<td>B</td><td>12</td><td>8D **</td><td>s₁: A, Q, D, H (rechtes Halbwort) oder B s₂: leer: + } statt ± : N : - s₃: leer: nicht speichern : C : zurückspeichern</td>}</td>}</td>}	<i> ± <s _{1<td>falls s₃ = C: <i> ± <s_{1<td>B</td><td>12</td><td>8D **</td><td>s₁: A, Q, D, H (rechtes Halbwort) oder B s₂: leer: + } statt ± : N : - s₃: leer: nicht speichern : C : zurückspeichern</td>}</td>}	falls s ₃ = C: <i> ± <s _{1<td>B</td><td>12</td><td>8D **</td><td>s₁: A, Q, D, H (rechtes Halbwort) oder B s₂: leer: + } statt ± : N : - s₃: leer: nicht speichern : C : zurückspeichern</td>}	B	12	8D **	s ₁ : A, Q, D, H (rechtes Halbwort) oder B s ₂ : leer: + } statt ± : N : - s ₃ : leer: nicht speichern : C : zurückspeichern
Modifiziere aus Speicher	MC	m	 := <m> + mod2 mod2 := <m> + mod2	sp	x		<m> *	<m> *		B	2	14	
Modifiziere aus Speicher nach Ersetzungen	MCE	m	 := <<...<m>...>> + mod2 mod2 := <<...<m>...>> + mod2	sp	x		<<...<m>...>> * das 1. Bit wird dem 2. angeglichen	<<...<m>...>> *		B	13x -9	17	x: Anzahl der Halbwörter in der Ersetzkette Abbruch wenn: <<...<m>...>> ₁ = L
Modifiziere mit Adressenteil	MA	z	 := z + mod2 mod2 := z + mod2	sp			z *	z *		B	1	03	z: 0...65 535
Modifiziere mit negativem Adressenteil	MNA	z	 := -z + mod2 mod2 := -z + mod2	sp			-z *	-z *		B	2	02	

* Ist vom vorhergehenden Befehl ein Modifikator
2. Art vorhanden, so wird er addiert

** siehe Internspezifikation auf Seite 23

Ersetzen (und modifizieren)

Bezeichnung	Code	adr	Wirkung	mod2	R	op	adr	(B)	(i)	mod2 _{1..n}	Werk	Take	Int.	Bemerkungen
<u>E</u> rsetze	E	c i	adr := <i> mod2 := mod2	sp		c	<i>			mod2	B	4	29	
<u>E</u> rsetze <u>z</u> ählend	EZ	c i	adr := <i> + 2 + mod2 := <i> + 2 <i> := <i> + 2	sp		c	<i> + 2 * <i> + 2	<i> + 2	<i> + 2	0	B	5	2B	
<u>E</u> rsetze <u>n</u> egativ <u>z</u> ählend	ENZ	c i	adr := <i> + mod2 := <i> - 2 <i> := <i> - 2	sp		c	<i> * <i> - 2	<i> - 2	<i> - 2	0	B	8	2A	
<u>E</u> rsetze und <u>m</u> odifiziere mit <u>B</u>	EMB	c i	adr := <i> mod2 := 			c	<i>				B	4	28	
<u>M</u> odifiziere <u>A</u> dressenteil mit <u>B</u>	MAB	c p	adr := <E> + p := + p			c	 + p + p	 + p		0	B	9,5	20	p: ±0...±127
<u>M</u> odifiziere <u>A</u> dressenteil mit <u>B</u> bei <u>I</u> nvarianz der <u>S</u> prungadresse	MABI	c p	adr := <E> + p := + p			c	 + p + p	 + p		0	F	9,5	3F	p: ±0...±127 **
<u>M</u> odifiziere über <u>U</u>	MU	c p	adr := <<U>> + p			c	<<U>> + p			0	B	14,5	05	<U> := <U>-1 wenn c Sprungbef. u. Bedingung erfüllt ist** p: ±0...±127
<u>E</u> rsetze nach <u>M</u> odifizierung über <u>U</u>	EMU	c p	adr := <<<U>> + p> mod2 := mod2	sp		c	<<<U>> + p>			mod2	B	29,5	04	p: ±0...±127
<u>R</u> elativ- <u>A</u> dressierung mit <u>R</u> egisterinhalt	RLR	c s	adr := <s> + <F> mod2 := mod2	sp		c	<s> + <F>	falls s = F: adr := 2·<F> + 1		mod2	B R	10	EO	s: A, Q, D, H = rechte Hälfte AL, QL, DL, HL = linke Hälfte F, B, U, Y
<u>R</u> egisteradressierung	R	c s	operand := <s ₁ >	+		c	<s ₁ > = Operand:	<div><div>s₁ = A, Q, D oder H: ro := t_{s₁}:<s₁>***</div><div>s₁ = Y, U : ro := 1; 0, <s₁></div><div>s₁ = B : ro := 1; v, </div><div>s₁ = F : ro := 1; v, <F>+1</div></div>	B R	4 6	96		s ₂ : L : linke 24 Bits : leer : rechte 24 Bits c : erlaubter Code ro: Registeroperand	
<u>T</u> ue	T	m	op, adr := <m> mod2 := mod2	sp	x	<m> ₁₋₈	0, <m> ₉₋₂₄			mod2	B	5	CC	<m> _t : beliebig

* Ist vom vorhergehenden Befehl ein Modifikator
2. Art vorhanden, so wird er addiert

** Sprung in eine andere Großseite ist möglich

***bei s₁ = A, Q, D oder H und <s₁>_t = 0 oder 1:
 <s₁>₁ := <s₁>_s
 <s₁>₂ := <s₁>_v

Als Zweitcodes (c) sind alle Befehls-
codes zugelassen.

Bei dem Befehl R jedoch nur:

A A2 AB AQ AT AU AUT

B B2 B2V B2VN B3 B3V BB BD BH BN
BNR BQ BGB BR BT BU

DV DVD DVI

ET

GA GAB GDV GDVI GMAN GML GMLA
GMLN GSB GSBB GSBD GSBI

HBC

M2 M2N M2NR M2R MAN MANR MAR
MC MCE MCF MCFU ML MLA MLD MLN
MLR MNR

NULL

REZ

SB SB2 SBB SBD SBI SBQ SBT SBU SE
SUE

T TCB

VBC VEL

ZUS

Aufbereitung

Bezeichnung	Code	adr	Wirkung	mod2	R	Voraussetzung	A	B	Q	Y	Werk	Takte	Int.	Bemerkungen
Umchlüsseln	US	s i	$\langle A \rangle := \langle A \rangle$ umgeschlüsselt	sp		$\langle A \rangle =$ umzu- schl. Zeich. $\langle i \rangle + \text{mod2} =$ Anfangsadr. der Umschl.- Tabelle *	$t_n; \langle A \rangle$ umge- schl. falls $s_2 = E$ $\langle A \rangle_{1-x} := 0$	$\langle i \rangle + \text{mod2}$ Adresse der Um- schl.- Tabelle	falls $s_2 = G$ $t_n; \langle A \rangle$ umge- schl.	falls $s_2 = G$ Anzahl der umgeschlüs- selt. Zeich.	B R	$s_2 = E: 32$ $s_2 = G: \text{und}$ $s_1 = 6: 176$ $= 8: 136$ $= C: 96$	E3	s_1 : Bitlänge der Zeichen 6, 8 oder C (12 Bits) s_2 : E = ein Zeichen rechtsbünd. G = alle Zeich. 8, 6 oder 4 t_n : Typenkennung des Wortes, in dem das linke Zeichen steht
Invertiere Register	IR	s	$\langle s \rangle := \neg \langle s \rangle$ $\langle s \rangle := \langle s \rangle $			falls s_2 : leer, invertiere falls s_2 : B, bilde Betrag im Register					R	$4p + 1$	E1	s_1 : Register A, Q, D und H p: Anzahl der adr. Register
Spezifikation														
BU Alarm														
Schifte	SH	s p	Schift gemäß Spezifika- tion um p Stellen	+		s_1 : A = Register A Q = Register Q AQ = Register A und Q getrennt Z = doppelt langes Register A, Q s_2 : leer, L = Rechtsschift, Linksschift s_3 : leer, K = gestreckter Schift, Kreisschift s_4 : leer, R = ohne Rundung, mit Rundung s_5 : leer, U = abhängig von TK, unabh. von TK s_6 : leer, B = nicht zählen, zählen der aus A geschifteten 'L'-Bits im Reg. Y	≥ 1				R	$2(q+r)+5$ p: 4=q Rest r zusätzl. 5 wenn $s_4=R$ u. gerundet wurde	9B	1) BU Alarm möglich: nur bei Linksschift in A mit TK=0. oder 1; bei Rechtsschift höchstens um 1 Stelle mit Rundung p= Anzahl der Schiftschritte $\pm 0 \dots +127$ p: 4 = q Rest r
Schifte in B	SHB	s p	$\langle B \rangle := \langle B \rangle$ geschiftet um p Stellen nach rechts oder links mit Nachziehen von 0-Bits			s: R = Schift nach rechts L = Schift nach links					B	$4 \cdot p + 6$	21	p: Anzahl der Schiftschritte 0...255
Alarm														
Y BU TK														
Vorzeichenangleich zwischen A und Q	VAQ	z	$\langle A, Q \rangle := \langle A \rangle + \langle Q \rangle \cdot 2^{-46}$ $\langle Q \rangle_v := \langle A \rangle_1$ falls $\langle A \rangle = \pm 0, \langle Q \rangle \neq \pm 0$ $\langle A \rangle_v := \langle Q \rangle_1$	+		VAQ gleicht die Vorzeichen an und berichtigt das Ergebnis unter Er- haltung des Zahlenwertes			$\langle A \rangle_t \neq 1$		R	2 bei $\langle A \rangle_1 = \langle Q \rangle_v$ 17 bei $\langle A \rangle_1 \neq \langle Q \rangle_v$	63	Adressenteil z ist ohne Bedeutung, muß aber an- gegeben werden
Normalisiere	NRM	s	bei s = G: $\langle A \rangle := \langle A \rangle$ normalisiert bei s = FG: $\langle A \rangle_{\text{einstk.}} := \langle A, Q \rangle_{\text{restk.}}$ bei s = F: $\langle A, Q \rangle := \langle A, Q \rangle$ normalisiert bei s = F4: $\langle A, Q \rangle := \langle A, Q \rangle$ normalisiert bei s = N falls $\langle A \rangle \neq +0$ $\langle A \rangle := \langle A \rangle$ links geschift. bis $\langle A \rangle_1 = L$ bei s = L: $\langle A \rangle := \langle A \rangle$ links geschift. bis $\langle A \rangle_1 = 0$	+		Die Festkommazahl wird als echter Bruch aufgefaßt $\langle A \rangle_t := 0; \langle A, Q \rangle$ gerundet, $\langle Q \rangle := 1; +0$ beim Schiften werden Nullen nachgezogen e falls $\langle A \rangle = +0$: $\langle A \rangle := \langle A \rangle$ beim Schiften werden Nullen nachgezogen e	$4t$ $+0$ $4t$ t 48 e	$\neq 0$ $\neq 1$ $\neq 1$ $\neq 1$		R	$2t + 3$ $2t + 10$ $2t + 7$ $2t + 7$ $2e + 3$	9F	s = G, FG, F, F4, N oder L t: Anzahl der Tetraden- schifte e: Anzahl der Einerschifte	

*Tabelle in Viertelwörtern.
Die Anfangsadresse ist eine Halbwortadresse

Bezeichnung	Code	adr	Wirkung	mod2	R	Voraussetzung	Y	BÜ	TK	Werk	Takte	Int.	Bemerkungen
Konvertiere Dezimalzahl in Festkommazahl rechtsbündig	KDFR	p	$\langle A \rangle_{\text{festk.}} := \langle A, Q \rangle_{\text{dez.}}$	+		$\langle A, Q \rangle = \text{pos. Dez.-Zahl Komma rechts}$ $\langle A \rangle := 1; \langle A, Q \rangle \langle D \rangle := 1; +0$ $\langle Q \rangle := 1; +0 \quad \langle H \rangle := 1; +0$	+0			R	7p + 22	94	p: 1...13 Anzahl der Dezimalstellen, die konvertiert werden
Konvertiere Festkommazahl linksbündig in Dezimalzahl	KFLD	p	$\langle A, Q \rangle_{\text{dez.}} := \langle A \rangle_{\text{festk.}}$	+		$\langle A \rangle = \text{pos. Festk.-Zahl Komma links}$ $\langle A \rangle_1 = 0 \text{ oder } 1$ $\langle A \rangle_1 = \langle A \rangle_2 = 0$	+0 \geq		2,3	R	$\langle A \rangle > 0$ 9p + 5	95	p: 1...13 Anzahl der gewünschten Dezimalstellen

* Tabelle in Viertelwörtern
Die Anfangsadresse ist eine Halbwortadresse

Tabelle durchsuchen

Suchbedingung			Eingangsgrößen $\langle B \rangle \quad \langle A \rangle \quad \langle Q \rangle \quad \langle D \rangle$			Alarm			BÜ TK					
Tabelle durchsuchen auf Identität	TLI	n	$\langle n+2k \rangle = \langle D \rangle$ ** Tabellenende: TK $\neq t_0$		$\langle D \rangle = \text{Suchwort}$ $\langle D \rangle_1 = \langle D \rangle_2$ bei $\langle D \rangle_1 = 0$ oder 1	n*	$t_0; \langle D \rangle$ ***	***	$\neq \langle D \rangle_t$	B R	15p + 10	EC	k: 0,1,2,... p: Anzahl der untersuchten Worte in der Tabelle x: 1...48 falls Dehnungswert $\langle B \rangle = \pm 0$ Wirkung wie Nullbefehl	
Tabelle durchsuchen mit Dehnung	TLD	n	$\langle n+k \langle B \rangle \rangle \geq \langle D \rangle$ ** Tabellenende: TK $\neq t_0$		$\langle D \rangle = \text{Suchwort}$ $\langle D \rangle_1 = \langle D \rangle_2$ bei $\langle D \rangle_1 = 0$ oder 1 $\langle B \rangle = \text{Dehnungswert}$	n*	$t_0; \langle D \rangle$ ***	***	$\neq \langle D \rangle_t$	B R	14,5p + 10,5	EB		
Tabelle durchsuchen mit Dehnung und Maske	TDM	n	$\langle n+k \langle B \rangle \rangle_x = \langle D \rangle_x$ ** für $\langle H \rangle_x = 0$ Tabellenende: TK $\neq t_0$		$\langle D \rangle = \text{Suchwort}$ $\langle H \rangle = \text{Maske}$ $\langle B \rangle = \text{Dehnungswert}$	n*	$t_H; \langle H \rangle \wedge \langle n^* \rangle$		$\neq \langle D \rangle_t$	B R	14,5p + 2,5	EA		
Tabelle durchsuchen auf Maximum	TMAX	n	$\langle n+k \langle B \rangle \rangle_x = \text{Max.}$ für $\langle H \rangle_x = 0$ Tabellenende: TK $\neq t_n$		$\langle H \rangle = \text{Maske}$ $\langle B \rangle = \text{Dehnungswert}$	n*	$t_n^*; \langle H \rangle \wedge \langle n^* \rangle$	$t_n^*; \neg \langle H \rangle \wedge \langle n^* \rangle$		B R	17p + 4	EF		
Tabelle durchsuchen auf Minimum	TMIN	n	$\langle n+k \langle B \rangle \rangle_x = \text{Min.}$ für $\langle H \rangle_x = 0$ Tabellenende: TK $\neq t_n$		$\langle H \rangle = \text{Maske}$ $\langle B \rangle = \text{Dehnungswert}$	n*	$t_n^*; \langle H \rangle \wedge \langle n^* \rangle$	$t_n^*; \neg \langle H \rangle \wedge \langle n^* \rangle$		B R	17p + 4	EE		
Tabelle durchsuchen logarithmisch	TLOG	n	$\langle n+2k \rangle_x \geq \langle D \rangle_x$ Voraussetzung: $\left. \begin{array}{l} \text{für } \langle H \rangle_x = 0 \\ \langle n \rangle_x < \langle D \rangle_x \end{array} \right\}$		$\langle A \rangle = \text{Tabellengänge in Ganzwört.}$ $2 \leq \langle A \rangle \leq 2^{20}$ $\langle D \rangle = \text{Suchwort}$ $\langle H \rangle = \text{Maske}$	n ⁺	$t_n^*; \langle H \rangle \wedge \langle n^+ \rangle$ falls kein Wort gefund. 3;0	$t_0; \neg \langle H \rangle \wedge \langle D \rangle$	kein Wort gef.	B R	23q + 46	ED	$\langle F \rangle := \langle F \rangle_{9-24} + 2$ falls $\langle n+2k \rangle_x > \langle D \rangle_x$ oder kein Wort gefunden q : $(2 \log \langle A \rangle) - 1 \leq q \leq 2 \log \langle A \rangle$ q : ganzzahlig	

* Adresse des zuerst gefundenen Wortes, das den Suchbedingungen genügt oder Adresse des ersten Wortes hinter der Tabelle (bei TMAX und TMIN immer Adresse des zuerst gefundenen Wortes)

** Wird kein Wort gefunden:
Abbruch mit TK-Alarm

*** falls $\langle D \rangle_t = 0$: $\langle D \rangle := \langle D \rangle$ normalisiert,
Vergleichsoperand normalisiert (im Speicher unverändert)

n⁺ niedrigste Adresse der nach den Suchbedingungen gefundenen Wörter oder erstes Wort hinter der Tabelle

Zentralcode

																<div> <div>2⁷</div> <div>1. Sede- zimale</div> </div> <div> <div>2⁰</div> <div>2. Sede- zimale</div> </div>	
0000	000L	00LO	00LL	0L00	0LOL	0LLO	0LLL	L000	L00L	LOLO	L0LL	LL00	LL0L	LLLO	LLLL		
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
⁰ NUL	¹⁶ NL6	³² SUB	⁴⁸	⁶⁴	⁸⁰	⁹⁶ "	¹¹² %	¹²⁸ ^	¹⁴⁴ +	¹⁶⁰ (¹⁷⁶ 0	¹⁹² A	²⁰⁸ Q	²²⁴ a	²⁴⁰ q	0	0000
¹	¹⁷ NL5	³³ EM	⁴⁹	⁶⁵	⁸¹	⁹⁷ ,	¹¹³ §	¹²⁹ v	¹⁴⁵ -	¹⁶¹)	¹⁷⁷ 1	¹⁹³ B	²⁰⁹ R	²²⁵ b	²⁴¹ r	1	000L
²	¹⁸ NL4	³⁴	⁵⁰	⁶⁶	⁸²	⁹⁸ /	¹¹⁴ #	¹³⁰ ¬	¹⁴⁶	¹⁶² [¹⁷⁸ 2	¹⁹⁴ C	²¹⁰ S	²²⁶ c	²⁴² s	2	00LO
³	¹⁹ NL3	³⁵ T1	⁵¹	⁶⁷	⁸³	⁹⁹ \	¹¹⁵ \$	¹³¹	¹⁴⁷ /	¹⁶³]	¹⁷⁹ 3	¹⁹⁵ D	²¹¹ T	²²⁷ d	²⁴³ t	3	00LL
⁴	²⁰ NL2	³⁶ T2	⁵²	⁶⁸	⁸⁴	¹⁰⁰	¹¹⁶ ¢	¹³²	¹⁴⁸	¹⁶⁴ {	¹⁸⁰ 4	¹⁹⁶ E	²¹² U	²²⁸ e	²⁴⁴ u	4	0L00
⁵	²¹ NL	³⁷ TE	⁵³ FL	⁶⁹	⁸⁵	¹⁰¹ ^	¹¹⁷	¹³³	¹⁴⁹	¹⁶⁵ }	¹⁸¹ 5	¹⁹⁷ F	²¹³ V	²²⁹ f	²⁴⁵ v	5	0LOL
⁶	²² CR	³⁸	⁵⁴	⁷⁰	⁸⁶	¹⁰² °	¹¹⁸ @	¹³⁴ ↑	¹⁵⁰	¹⁶⁶ <	¹⁸² 6	¹⁹⁸ G	²¹⁴ W	²³⁰ g	²⁴⁶ w	6	0LLO
⁷	²³ NF	³⁹	⁵⁵	⁷¹	⁸⁷	¹⁰³ ~	¹¹⁹ &	¹³⁵	¹⁵¹ =	¹⁶⁷ >	¹⁸³ 7	¹⁹⁹ H	²¹⁵ X	²³¹ h	²⁴⁷ x	7	0LLL
⁸	²⁴ VT	⁴⁰	⁵⁶	⁷²	⁸⁸	¹⁰⁴	¹²⁰ *	¹³⁶	¹⁵² ≠	¹⁶⁸ MZ	¹⁸⁴ 8	²⁰⁰ I	²¹⁶ Y	²³² i	²⁴⁸ y	8	L000
⁹	²⁵ VT3	⁴¹	⁵⁷	⁷³	⁸⁹	¹⁰⁵	¹²¹	¹³⁷	¹⁵³	¹⁶⁹ .	¹⁸⁵ 9	²⁰¹ J	²¹⁷ Z	²³³ j	²⁴⁹ z	9	L00L
¹⁰	²⁶ VT4	⁴²	⁵⁸	⁷⁴	⁹⁰	¹⁰⁶	¹²²	¹³⁸	¹⁵⁴	¹⁷⁰ ,	¹⁸⁶	²⁰² K	²¹⁸ Ä	²³⁴ k	²⁵⁰ ä	A	LOLO
¹¹	²⁷ VT5	⁴³	⁵⁹	⁷⁵	⁹¹	¹⁰⁷	¹²³	¹³⁹	¹⁵⁵ <	¹⁷¹ :	¹⁸⁷	²⁰³ L	²¹⁹ Ö	²³⁵ l	²⁵¹ ö	B	L0LL
¹²	²⁸ VT6	⁴⁴	⁶⁰	⁷⁶	⁹²	¹⁰⁸ ´	¹²⁴ ¤	¹⁴⁰	¹⁵⁶ >	¹⁷² ;	¹⁸⁸	²⁰⁴ M	²²⁰ Ü	²³⁶ m	²⁵² ü	C	LL00
¹³	²⁹ VT7	⁴⁵	⁶¹	⁷⁷	⁹³	¹⁰⁹ `	¹²⁵	¹⁴¹ 10	¹⁵⁷ ≡	¹⁷³ !	¹⁸⁹	²⁰⁵ N	²²¹	²³⁷ n	²⁵³ ß	D	LL0L
¹⁴	³⁰ VT8	⁴⁶	⁶²	⁷⁸	⁹⁴	¹¹⁰ -	¹²⁶	¹⁴²	¹⁵⁸ ≡	¹⁷⁴ ?	¹⁹⁰	²⁰⁶ O	²²²	²³⁸ o	²⁵⁴	E	LLLO
¹⁵	³¹ NL7	⁴⁷	⁶³	⁷⁹	⁹⁵	¹¹¹ -	¹²⁷ π	¹⁴³	¹⁵⁹	¹⁷⁵ SP	¹⁹¹ PZ	²⁰⁷ P	²²³	²³⁹ p	²⁵⁵	F	LLLL
Steuerzeichen						Akzente	Sonstige Zeichen	Mathematische Zeichen	Satzzeichen	Ziffern	Großbuchstaben	Kleinbuchstaben				GR 24	

CR Wagenrücklauf (Carriage Return)
 EM Ende der Aufzeichnung (End of Medium)
 FL Fluchtsymbol
 MZ Minus Null (Minus Zero)

NF Neues Formular (New Form)
 NL Neue Zeile (New Line)
 NLx Varschub um x Zeilen
 NUL Nil (Null)

PZ Plus Null (Plus Zero)
 SP Zwischenraum (Space)
 SUB Substitution (Substitute Character)
 T1 Textumschaltung 1

T2 Textumschaltung 2
 TE Textende
 VT Vertikal-Tabulator (Vertical Tabulation)
 VTx Zeilenvorschub gemäß 5pur x

Potenzen von 2

2 ⁿ	n	2 ⁻ⁿ	2 ⁿ	n	2 ⁻ⁿ
2	1	0.5	2 251 799 813 685 248	51	0.(15) 444 089 209 850 062 616 169 452 667 236 328 125
4	2	0.25	4 503 599 627 370 496	52	0. 222 044 604 925 031 306 084 726 333 618 164 062 5
8	3	0.125	9 007 199 254 740 992	53	0. 111 022 302 462 515 654 042 363 166 809 082 031 25
16	4	0.062 5	18 014 398 509 481 984	54	0. 055 511 151 231 257 827 021 181 583 404 541 015 625
32	5	0.031 25	36 028 797 018 963 968	55	0. 027 755 575 615 628 913 510 590 781 702 270 507 812 5
64	6	0.015 625	72 057 594 037 927 936	56	0.(15) 013 877 787 807 814 456 755 295 395 851 135 253 906 25
128	7	0.007 812 5	144 115 188 075 855 872	57	0. 006 938 893 903 907 228 377 647 697 925 567 626 853 125
256	8	0.003 906 25	288 230 376 151 711 744	58	0. 003 469 446 951 953 614 188 823 848 962 783 813 478 562 5
512	9	0.001 953 125	576 460 752 303 423 488	59	0. 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 024	10	0.000 876 562 5	1 152 921 504 606 846 976	60	0. 000 867 361 737 988 403 547 205 962 240 695 953 389 140 625
2 048	11	0.000 488 281 25	2 305 843 009 213 693 852	61	0.(18) 433 680 868 994 201 773 602 981 120 347 876 684 570 312 5
4 096	12	0. 244 140 625	4 611 688 018 427 387 904	62	0. 216 840 434 497 100 886 901 490 560 173 888 342 285 156 25
8 192	13	0. 122 070 312 5	9 223 372 036 854 775 808	63	0. 108 420 217 248 550 443 400 745 280 086 894 171 142 578 125
16 384	14	0. 061 035 156 25	18 446 744 073 709 551 616	64	0. 054 210 108 624 275 221 700 372 640 043 497 085 571 289 062 5
32 768	15	0. 030 517 578 125	36 893 488 147 419 103 232	65	0. 027 105 054 312 137 610 850 186 320 021 748 542 785 644 531 25
65 536	16	0.000 015 258 789 062 5	73 786 976 294 838 206 464	66	0.(18) 013 552 527 156 068 805 425 093 160 010 874 271 392 822 265 625
131 072	17	0. 007 629 394 531 25	147 573 952 589 676 412 928	67	0. 006 776 263 578 034 402 712 546 580 005 437 135 696 411 132 812 5
262 144	18	0. 003 814 697 265 625	295 147 905 179 352 825 856	68	0. 003 388 131 789 017 201 356 273 290 002 718 567 848 205 566 406 25
524 288	19	0. 001 907 348 632 812 5	590 295 810 358 705 851 712	69	0. 001 694 065 894 508 600 678 136 645 001 359 283 924 102 783 203 125
1 048 576	20	0. 000 953 674 316 406 25	1 180 591 620 717 411 303 424	70	0. 000 847 032 947 254 300 339 068 322 500 679 841 962 051 391 601 562 5
2 097 152	21	0.000 000 476 837 158 203 125	2 361 183 241 434 822 606 848	71	0.(21) 423 516 473 627 150 169 534 161 250 339 820 981 025 695 800 781 25
4 194 304	22	0. 238 418 579 101 562 5	4 722 366 482 869 645 213 696	72	0. 211 758 236 813 575 084 767 080 625 169 910 490 512 847 900 390 625
8 388 608	23	0. 119 209 289 550 781 25	9 444 732 965 739 290 427 392	73	0. 105 879 118 406 787 542 383 540 312 584 955 245 256 423 950 195 312 5
16 777 216	24	0. 059 604 644 775 390 625	18 889 465 831 478 580 854 784	74	0. 052 939 559 203 393 771 191 770 156 292 477 622 628 211 975 097 656 25
33 554 432	25	0. 029 802 322 387 696 312 5	37 778 931 862 957 161 709 568	75	0. 026 468 779 801 696 885 595 885 078 146 238 811 314 105 987 548 828 125
67 108 864	26	0.000 000 014 901 161 193 847 656 25	75 557 863 725 914 323 418 136	76	0.(21) 013 234 889 800 848 442 797 942 539 073 118 405 657 052 993 774 414 062 5
134 217 728	27	0. 007 450 580 596 923 828 125	151 115 727 451 828 648 838 272	77	0. 006 617 444 900 424 221 398 971 269 538 559 702 828 528 496 887 207 031 25
268 435 456	28	0. 003 725 290 298 481 914 062 5	302 231 454 903 657 293 676 544	78	0. 003 308 722 450 212 110 699 485 634 768 278 851 414 283 248 443 603 515 625
536 870 912	28	0. 001 862 645 149 230 957 031 25	604 462 909 807 314 587 353 088	78	0. 001 854 361 225 106 055 349 742 817 384 139 825 707 131 624 221 801 757 812 5
1 073 741 824	30	0. 000 931 322 574 815 478 515 625	1 208 825 819 614 629 174 706 176	80	0. 000 827 180 612 553 027 674 871 408 692 069 962 853 565 812 110 900 878 906 25
2 147 483 648	31	0.000 000 000 465 861 287 307 738 257 812 5	2 417 851 639 229 258 349 412 352	81	0.(24) 413 590 306 276 513 837 435 704 346 034 981 426 782 906 055 450 439 453 125
4 294 967 296	32	0. 232 830 643 653 869 628 906 25	4 835 703 278 458 516 688 824 704	82	0. 206 795 153 138 256 918 717 852 173 017 490 713 391 453 027 725 218 726 562 5
8 589 934 592	33	0. 116 415 321 826 934 814 453 125	9 671 406 556 917 033 397 649 408	83	0. 103 397 576 569 128 459 358 826 088 508 745 356 695 728 513 862 609 863 281 25
17 179 869 184	34	0. 058 207 660 813 467 407 228 582 5	19 342 813 113 834 066 795 298 818	84	0. 051 698 788 284 564 229 679 463 043 254 372 678 347 863 256 931 304 931 640 625
34 359 738 368	35	0. 029 103 830 458 733 603 613 281 25	38 685 626 227 668 133 590 597 632	85	0. 025 848 394 142 282 114 839 731 521 627 186 339 173 931 628 465 652 465 820 312 5
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625	77 371 252 455 336 267 181 195 264	86	0.(24) 012 924 697 071 141 057 419 865 760 813 593 169 586 865 814 232 826 232 810 156 25
137 438 953 472	37	0. 007 275 957 614 183 425 903 320 312 5	154 742 504 910 672 534 362 390 528	87	0. 006 462 348 535 570 528 709 932 880 406 796 584 793 482 907 118 413 116 455 078 125
274 877 906 944	38	0. 003 637 878 807 091 712 851 660 156 25	309 485 009 821 345 068 724 781 056	88	0. 003 231 174 267 785 264 354 966 440 203 398 292 396 741 453 558 206 558 227 539 062 5
549 755 813 888	39	0. 001 818 989 403 545 856 475 830 078 125	618 870 019 842 690 137 449 562 112	88	0. 001 615 587 133 892 632 177 483 220 101 699 146 198 370 726 779 103 279 113 769 531 25
1 099 511 627 776	40	0. 000 909 494 701 772 928 237 915 038 062 5	1 237 940 039 285 380 274 899 124 224	90	0. 000 807 793 568 946 316 068 741 610 050 849 573 099 185 363 389 551 639 556 884 765 625
2 199 023 255 552	41	0.000 000 000 000 454 747 350 888 464 118 957 519 531 25	2 475 880 078 570 760 549 798 248 448	81	0.(27) 403 896 783 473 158 044 370 805 025 424 786 549 592 881 694 775 819 778 442 382 812 5
4 398 046 511 104	42	0. 227 373 675 443 232 069 478 759 765 625	4 951 760 157 141 521 099 596 496 896	92	0. 201 948 391 738 579 022 185 402 512 712 393 274 796 340 847 387 909 889 221 181 406 25
8 796 093 022 208	43	0. 113 686 837 721 816 029 739 378 882 812 5	9 903 520 314 283 042 199 192 993 792	93	0. 100 974 195 868 289 511 092 701 256 356 196 637 398 170 423 693 954 610 595 703 125
17 592 186 044 416	44	0. 056 843 418 860 808 014 869 889 941 406 25	19 807 040 628 566 084 398 385 987 584	94	0. 050 487 097 934 144 755 546 350 628 178 098 318 699 085 211 846 877 472 305 297 851 562 5
35 184 372 088 832	45	0. 028 421 709 430 404 007 434 844 970 703 125	39 614 081 257 132 188 796 771 975 168	95	0. 025 243 548 967 072 377 773 175 314 089 049 159 349 542 605 823 488 736 152 648 925 781 25
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5	79 228 162 514 264 337 593 543 950 336	98	0.(27) 012 621 774 483 536 188 896 587 657 044 524 579 674 771 302 961 744 368 076 324 462 890 825
140 737 488 355 328	47	0. 007 105 427 357 601 001 858 711 242 675 781 25	158 456 325 028 528 675 187 087 900 672	97	0. 006 310 887 241 768 094 443 293 828 522 262 289 837 385 651 480 872 184 038 162 231 445 312 5
281 474 976 710 656	48	0. 003 552 713 678 800 500 829 355 621 337 890 625	318 912 650 057 057 350 374 175 801 344	98	0. 003 155 443 620 884 047 221 646 914 261 131 144 918 692 825 740 436 092 018 081 115 722 656 25
562 949 953 421 312	48	0. 001 776 356 839 400 250 464 677 810 668 945 312 5	633 825 300 114 114 700 748 351 602 688	99	0. 001 577 721 810 442 023 810 823 457 130 565 572 459 346 412 870 218 046 009 540 557 861 328 125
1 125 899 906 842 624	50	0. 000 888 178 419 700 125 232 338 905 334 472 858 25	1 267 850 600 228 229 401 496 703 205 376	100	0. 000 788 860 905 221 011 805 411 728 565 282 786 229 673 206 435 109 023 004 770 278 930 664 062 5

Konvertierungstafel

sedezimal		dezimal									
100 000	1 048 576	010 000	65 536	001 000	4 096	000 100	256	000 010	16	000 001	1
200 000	2 097 152	020 000	131 072	002 000	8 192	000 200	512	000 020	32	000 002	2
300 000	3 145 728	030 000	196 608	003 000	12 288	000 300	768	000 030	48	000 003	3
400 000	4 194 304	040 000	262 144	004 000	16 384	000 400	1 024	000 040	64	000 004	4
		050 000	327 680	005 000	20 480	000 500	1 280	000 050	80	000 005	5
		060 000	393 216	006 000	24 576	000 600	1 536	000 060	96	000 006	6
		070 000	458 752	007 000	28 672	000 700	1 792	000 070	112	000 007	7
		080 000	524 288	008 000	32 768	000 800	2 048	000 080	128	000 008	8
		090 000	589 824	009 000	36 864	000 900	2 304	000 090	144	000 009	9
		0A0 000	655 360	00A 000	40 960	000 A00	2 560	000 0A0	160	000 00A	10
		0B0 000	720 896	00B 000	45 056	000 B00	2 816	000 0B0	176	000 00B	11
		0C0 000	786 432	00C 000	49 152	000 C00	3 072	000 0C0	192	000 00C	12
		0D0 000	851 968	00D 000	53 248	000 D00	3 328	000 0D0	208	000 00D	13
		0E0 000	917 504	00E 000	57 344	000 E00	3 584	000 0E0	224	000 00E	14
		0F0 000	983 040	00F 000	61 440	000 F00	3 840	000 0F0	240	000 00F	15
dezimal		sedezimal									
1 000 000	0F4 240	100 000	018 6A0	10 000	002 710	1 000	000 3E8	100	000 064	10	000 00A
2 000 000	1E8 480	200 000	030 D40	20 000	004 E20	2 000	000 7D0	200	000 0C8	20	000 014
3 000 000	2DC 6C0	300 000	049 3E0	30 000	007 530	3 000	000 BB8	300	000 12C	30	000 01E
4 000 000	3D0 900	400 000	061 A80	40 000	009 C40	4 000	000 FA0	400	000 190	40	000 028
		500 000	07A 120	50 000	00C 350	5 000	001 388	500	000 1F4	50	000 032
		600 000	092 7C0	60 000	00E A60	6 000	001 770	600	000 258	60	000 03C
		700 000	0AA E60	70 000	011 170	7 000	001 858	700	000 28C	70	000 046
		800 000	0C3 500	80 000	013 880	8 000	001 F40	800	000 320	80	000 050
		900 000	0D8 8A0	90 000	015 F90	9 000	002 328	900	000 384	90	000 05A

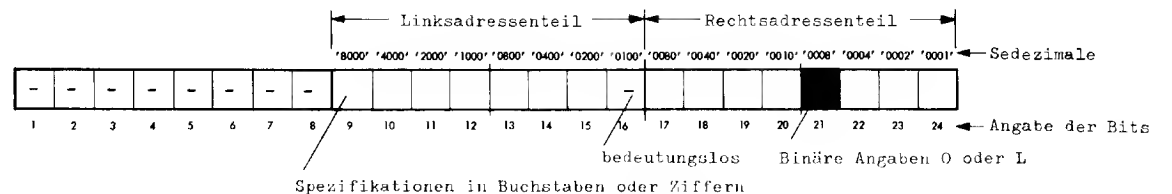
Befehlscode Intern - Extern

0000	000L	00LO	00LL	0L00	0LOL	0LL0	0LLL	L000	L00L	LOLO	L0LL	LL00	LL0L	LLLO	LLLL		
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
NULL	LZL	MAB	ZMC	AB		DV	B	C	ST	C2	BL		CU	RLR	DA	0	0000
XBA	HBA	SHB	LMC	SBB		DVD	BD	CR	STN	C3	VPU *		BZN	IR	DSB	1	000L
MNA	NL	WTV	LMT	A	GAB	DVI	BQ	CMT	ZTR	CMC	SXR		ZDP *		DML	2	00LO
MA	VBA	WTR	LC	AC	GSBB	VAQ	BH	CMR	SEGG	BC	SXRN		BU	US	MLD	3	00LL
EMU	MC	SXI	SM	SBI	ML	GDV	BB	CN	KDFR	SI0	Y *		SG0		AT	4	0L00
MU	VBC	SXGG	SMN	SBD	MLR	REZ	BN	CB	KFLD	SKG0	LEI *		SK0		SBT	5	0LOL
BCL	MCF	SXKG	S	SB	MLA	GDVI	BR	CD	R	SGG0	BCI			BNZ	BT	6	0LL0
TBC	MCE	SXN	VSS *	SBC	MAR	B2VN	BNR	CQ	RT	SN0	ZI			CNZ	CT	7	0LLL
MFU** M, XB	XC** XCN	EMB	SU	GSBI	MLN	VEL	M2N	VLA	AA	SAT	SR	ZT0	BZ2		WB	8	L000
MD	XBAN	E	TCB	AU	MNR	AUT	M2NR	ATA	SBA	SAA	PDP	ZT1	BZ		SBIT	9	L00L
SZX	ZX	ENZ	SFB	GAC	MAN	ET	M2	ETA	LR	SK	SRN	ZT2	BQB	TDM	SFBE	A	LOLO
MF	SW *	EZ		GA	MANR	ZUS	M2R	LA	SH	SG	SSR	ZT3	CZ	TLD	BSS	B	L0LL
TXX	SLN	HXP	HBC	GSBD	GMLN	B3	A2	TXR	TRX	SI	SE	T	BAR	TLI	ZK	C	LL00
TTX	SNL	MH	MCFU	SBU	GMAN	B3V	SB2	RX** MRX	HALT *	SN	SUE		BANR	TLOG	TOK	D	LL0L
MHX	SL	HXX	ZU	GSBC	GML	B2	AQ	BA		SKG	BLEI	SXG		TMIN	QBR	E	LLL0
HBPX	SLL	VXX	MABI	GSB	GMLA	B2V	SBQ	CH	NRM	SGG	VMO *	SXK	BAN	TMAX	QCR	F	LLLL

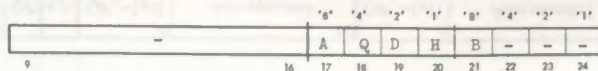
OK

*nicht für die Programmierung von Operatoren. Der Befehl SSR findet in Abwicklerdiensten Verwendung. **Unterscheidung im Adressenteil (siehe Internspezifikationen)

Spezifikationen - Intern

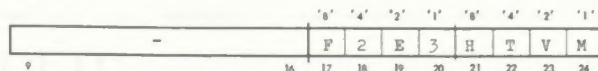


IR s



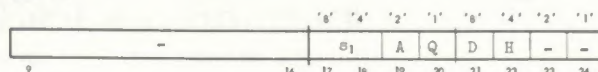
Es sind alle 2⁸ möglichen Bitsordnungen erlaubt. Falls die Bits 17 - 20 = 0; Wirkung wie Nullbefehl.

LA s



s: F, 2, E, 3, V und M können beliebig kombiniert werden. H oder T dürfen nur einzeln oder mit M verwendet werden. Nicht erlaubte Spezifikationen ergeben undefinierte Befehlsausführungen.

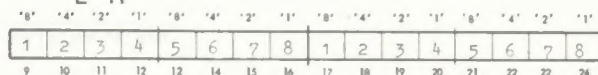
LR s



Es sind alle 2⁸ möglichen Bitsordnungen erlaubt.

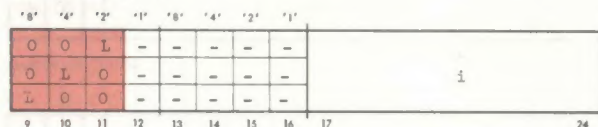
s₁: Typenennung 0,1,2 oder 3

LZL s_L s_R



Es sind in jedem Adressenteil jeweils alle 2⁸ möglichen Bitsordnungen erlaubt.

MFU, XB, M i

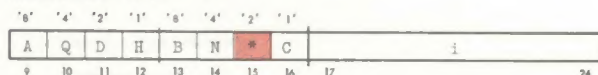


MFU: Bits 9 - 11 (Linkssadresse) = OOL

XB: Bits 9 - 11 (Linkssadresse) = OLO

M: Bits 9 - 11 (Linkssadresse) = LOO

MRX, RX s i



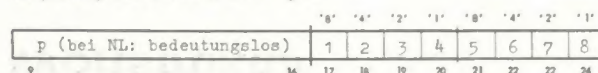
*MRX: Bit 15 = L

RX: Bit 15 = 0

Von den Bits 9 - 13 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

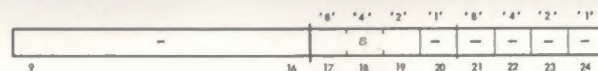
NL s

SL, SLL, SLN, SNL, SW p s



Es sind alle 2⁸ möglichen Bitsordnungen erlaubt.

NRM s

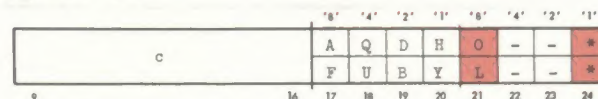


s besteht nur aus den Bits 17 - 19

N	O	O	O
L	O	O	L
F	O	L	O
F4	O	L	L
G	L	O	O
**	L	O	L
FG	L	L	O
**	L	L	L

** diese Bitanordnungen führen zu undefinierten Befehlsausführungen.

R, RLR c s

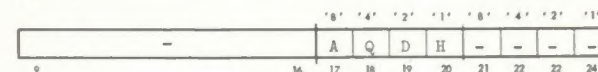


Gensu eines der Bits 17 - 20 muß = L sein, sonst undefinierte Befehlsausführung. (Ausnahme: Falls c = Null, so ist Bit 17 - 20 = 0 erlaubt)

* Bit 24 = L rechte 24 Bits } nur bei Halbwortbefehlen

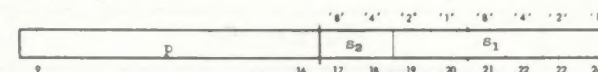
* Bit 24 = 0 linke 24 Bits } von Bedeutung

RT s



Gensu zwei der Bits 17 - 20 müssen = L sein, sonst undefinierte Befehlsausführung.

SBIT p s

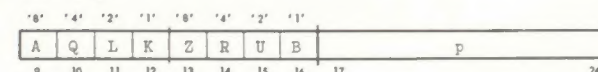


s₂: Register

A	O	O
Q	O	L
D	L	O
H	L	L

s₁: Dualziffern 1-48

SH s p



Es sind alle 2⁸ möglichen Bitsordnungen erlaubt.

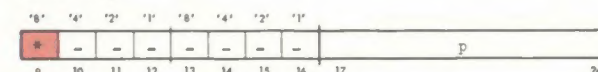
Wenn Bit 13 = L, sind die Bits 9 und 10 bedeutungslos.

Wenn Bit 12 = L, ist Bit 14 bedeutungslos.

Wenn Bit 11 = L und Bit 13 = 0, ist Bit 14 bedeutungslos.

Wenn Bit 9,10 und 13 = 0, so sind die Bits 11,12,14 und 15 bedeutungslos.

SHB s p

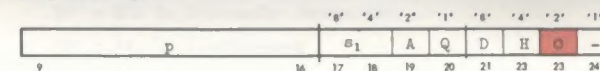


Es sind alle 2⁸ möglichen Bitsordnungen erlaubt

* Bit 9 = L: links

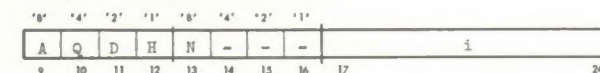
Bit 9 = 0: rechts

ST, STN p s



Von den Bits 19 - 22 darf nur eines = L sein, sonst erfolgt eine undefinierte Befehlsausführung. Bit 23 muß = 0 sein, sonst wird in jedem Fall gesprungen.

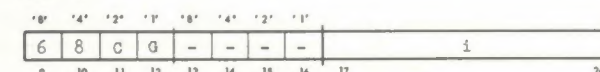
TRX, TXR s i



TRX: Von den Bits 9 - 12 darf nur eines = L sein, sonst undefinierte Befehlsausführung.

TXR: Es sind alle 2⁸ möglichen Bitanordnungen erlaubt.

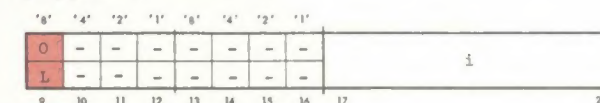
US s i



Von den Bits 9 - 11 muß gensu eines = L sein, sonst undefinierte Befehlsausführung.

Wenn das Bit 12 = 0, dann Teilspezifikation E.

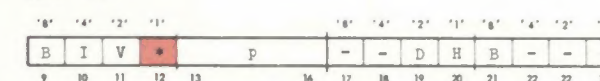
XC, XCN i



XC: Bit 9 (Linkssadresse) = 0

XCN: Bit 9 (Linkssadresse) = L

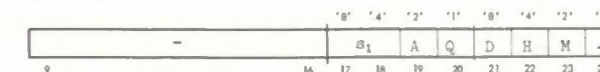
ZK s i



* Wenn Bit 12 = L(R), muß gensu eines der Bits 19-21 = L sein, sonst undefinierte Befehlsausführung

Wenn Bit 12 = 0(leer), wird in den Bits 17-24 die Adresse einer Indexzelle erwartet

ZTR s

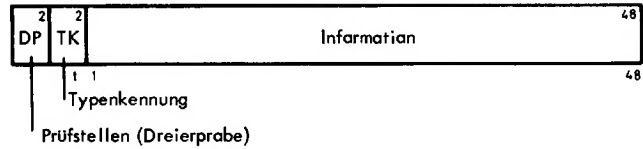


s₁: Typenennung 0,1,2 oder 3

Von den Bits 19 - 22 darf höchstens eines = L sein, sonst undefinierte Befehlsausführung.

Wortstruktur (im Speicher)

ALLGEMEIN

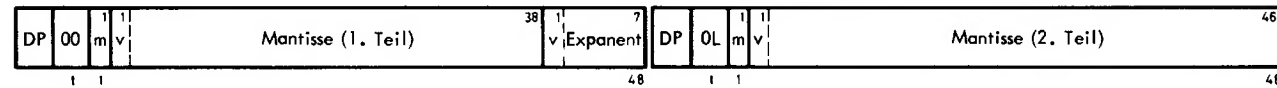


GLEITKOMMAZAHL (Basis 16)

Einfache Länge

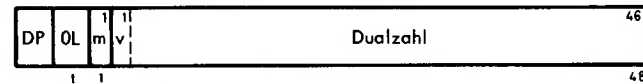


Doppelte Länge

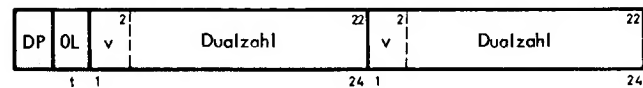


FESTKOMMAZAHL

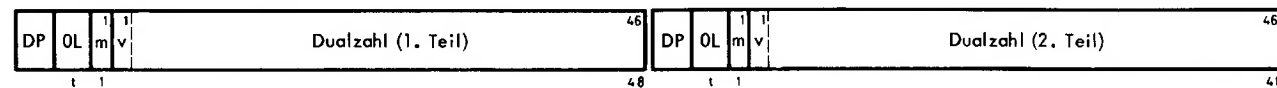
Einfache Länge



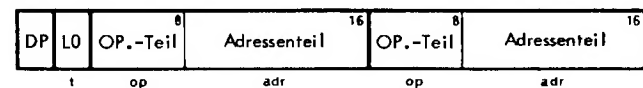
Halbe Länge (zwei Zahlen pro Wort)

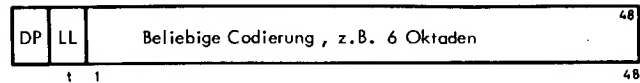


Doppelte Länge



BEFEHLE (zwei Befehle pro Wort)





ER 1711

DP = Bits für Prüfzwecke (Dreierprobe)

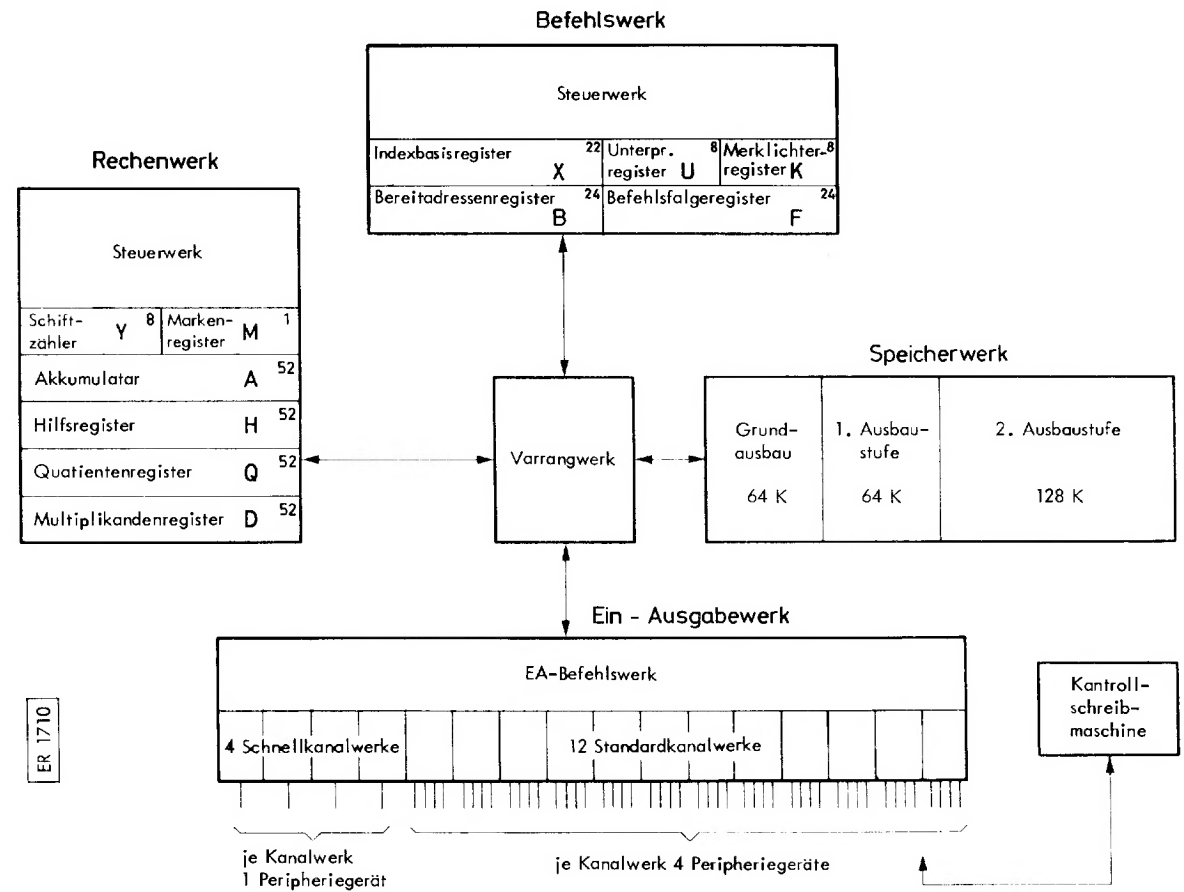
TK = Bits für Typenkennung

t = Typenkennung

m = Marke (nur im Speicher, im Register gleich der v-Stelle)

v = Vorzeichen

Blockschaltbild



ER 1710

Alphabetische Liste der Befehle

Code	Int.	Seite	mod2	R	Op	Code	Int.	Seite	mod2	R	Op	Code	Int.	Seite	mod2	R	Op	Code	Int.	Seite	mod2	R	Op	Code	Int.	Seite	mod2	R	Op
A	42	7	+	x	+	CN	84	5	+			LA	8B	10	+			S	36	12	+			SXRN	B3	13			
A2	7C	9	+	x	+	CNZ	E7	5	sp			LC	33	11	+	+		SAA	A9	13	+			SZX	0A	13			
AA	98	78	+			CQ	87	5	+			LMC	31	11	+	+		SAT	A8	13	+								
AB	40	7	+	x	+	CR	81	5	+			LMT	32	11	+	+		SB	46	7	+	x	+						
AC	43	7	+	+	+	CT	F7	5	+	+		LR	9A	10	+			SB2	7D	9	+	x	+	T	CC	15	sp	x	+
AQ	7E	7	+	x	+	CU	D0	5	+			LZL	10	11				SBA	99	78	+			TBC	07	6		+	+
AT	F4	9	+	x	+	CZ	DB	5	+									SBB	41	7	+	x	+	TCB	39	6	+	x	+
ATA	89	9	+									M	08	14	sp			SBC	47	7	+	+	+	TDM	EA	17		+	+
AU	49	8	+	x	+	DA	F0	8		+		M2	7A	9	+	x	+	SBD	45	7	+	x	+	TLD	EB	17		+	
AUT	69	9	+	x	+	DML	F2	8		+		M2N	78	9	+	x	+	SBI	44	7	+	x	+	TLI	EC	17		+	
						DSB	F1	8		+		M2NR	79	9	+	x	+	SBIT	F9	13				TLOG	ED	17		+	
B	70	4	+	x	+	DV	60	7	+	x	+	M2R	7B	9	+	x	+	SBQ	7F	7	+	x		TMAX	EF	17		+	
B2	6E	4	+	x	+	DVD	61	7	+	x	+	MA	03	14	sp			SBT	F5	9	+	x	+	TMIN	EE	17		+	
B2V	6F	4	+	x	+	DVI	62	7	+	x	+	MAB	20	15				SBU	4D	8	+	x	+	TOK	FD	6			
B2VN	67	4	+	x	+							MABI	3F	15				SE	BC	12	sp	x	+	TRX	9C	6			
B3	6C	4	+	x	+							MAN	5A	7	+	x	+	SEGG	93	13	+			TTX	0D	6			
B3V	6D	4	+	x	+							MANR	5B	7	+	x	+	SFB	3A	12	+			TXR	8C	6			
BA	8E	10	+			E	29	15	sp			MAR	57	7	+	x	+	SFBE	FA	12	sp			TXX	0C	6			
BAN	DF	10	+			EMB	28	15				MC	14	14	sp	x	+	SG	AB	12	+								
BANR	DD	10	+			EMU	04	15	sp			MCE	17	14	sp	x	+	SG0	D4	12	+			US	E3	16	sp		
BAR	DC	10	+			ENZ	2A	15	sp			MCF	16	14	sp	x	+	SGG	AF	12	+								
BB	74	4	+	x	+	ET	6A	9	+	x	+	MCFU	3D	14	sp	x	+	SGG0	A6	12	+			VAQ	63	16	+		
BC	A3	5	+			ETA	8A	9	+			MD	09	14	sp			SH	9B	16	+			VBA	13	10			
BCI	B6	5				EZ	2B	15	sp			MF	0B	14	sp			SHB	21	16				VBC	15	10		x	+
BCL	06	6										MFU	08	14	sp			SI	AC	12	+			VEL	68	9	+	x	+
BD	71	4	+	x	+	GA	4B	8	+	x	+	MH	2D	14				SIO	A4	12	+			VLA	88	9	+		
BH	73	4	+	x	+	GAB	52	8	+	x	+	MHX	0E	14				SK	AA	12	+			VXX	2F	10			
BL	80	5	+			GAC	4A	8	+	+		ML	54	7	+	x	+	SK0	D5	12	+								
BLEI	BE	4	+			GDV	64	8	+	x	+	MLA	56	7	+	x	+	SKG	AE	12	+			WB	F8	12			
BN	75	4	+	x	+	GDVI	66	8	+	x	+	MLD	F3	8				SKG0	A5	12	+			WTR	23	6			
BNR	77	4	+	x	+	GMAN	5D	8	+	x	+	MLN	58	7	+	x	+	SL	1E	13				WTV	22	6			
BNZ	E6	4	sp			GML	5E	8	+	x	+	MLR	55	7	+	x	+	SLL	1F	13									
BQ	72	4	+	x	+	GMLA	5F	8	+	x	+	MNA	02	14	sp			SLN	1C	13									
BQB	DA	4	+	x	+	GMLN	5C	8	+	x	+	MNR	59	7	+	x	+	SM	34	13	+								
BR	76	4	+	x	+	GSB	4F	8	+	x	+	MRX	8D	14				SMN	35	13	+								
BSS	FB	4				GSBB	53	8	+	x	+	MU	05	15				SN	AD	12	+								
BT	F6	4	+	x	+	GSBC	4E	8	+	+								SNO	A7	12	+								
BU	D3	4	+	x	+	GSBD	4C	8	+	x	+	NL	12	11				SNL	1D	13									
BZ	D9	4	+			GSBI	48	8	+	x	+	NRM	9F	16	+			SR	B8	12	+								
BZ2	D8	4	+	+								NULL	00	12		x		SRN	BA	12	+								
BZN	D1	4	+															SSR	BB	12	+								
						HBA	11	10				PDP	B9	13	+	+		ST	90	13	+								
C	80	5	+			HBC	3C	10		x	+							STN	91	13	+								
C2	A0	5	+	+		HBPX	0F	10										SU	38	12	+								
C3	A1	5	+	+		HXP	2C	10				QBR	FE	6		+		SUE	BD	12	sp	x	+						
CB	85	5	+			HXX	2E	10				QCR	FF	6				SXG	CE	13									
CD	86	5	+															SXGG	25	13									
CH	8F	5	+			IR	E1	16				R	96	15	+			SXI	24	13									
CMC	A2	5	+	+								REZ	65	8	+	x	+	SXK	CF	13									
CMR	83	5	+			KDFR	94	17	+			RLR	E0	15	sp			SXKG	26	13									
CMT	82	5	+			KFLD	95	17	+			RT	97	6	+			SXN	27	13									
												RX	8D	10				SXR	B2	13									

Code: Befehlscode

Int.: Interncode in 2 Sedezimalen

mod2: + = Modifizierung 2. Art

mod2: sp = spezielle Modifizierung 2. Art

R: x = als Zweitcode beim Befehl R zugelassen

Op: Dieser Befehl holt in der Abrufphase einen Operanden aus dem Speicher